英特尔中国
医疗健康行业
AI实战手册

intel
XEON

Contents

目录

趋势篇

实战篇

技术篇

硬件产品

软件和框架

注：* 部分为 2023 年版本更新内容

趋势篇

人工智能在医疗健康领域的发展与应用

人工智能在医疗健康领域的发展

医疗人工智能的市场趋势

得益于算法的进一步成熟、算力的提高以及数据的持续积累，人工智能（Artificial Intelligence，AI）得到迅猛发展，深度学习成为其代表，并呈现出应用领域日益集中的趋势。

作为 AI 技术最重要的落地领域之一，医疗行业与人工智能技术的结合也在近年来获得了市场的巨大青睐。据弗若斯特沙利文（Frost & Sullivan）发布的研究报告显示，中国医疗智能行业市场规模正在呈现高速增长，预计将在 2030 年超过 1.1 万亿元人民币[1]。这一高速增长一方面得益于中国医疗市场的迫切需求，另一方面则源于近年来医疗人工智能技术的发展以及相关政策的支持。同时，人工智能技术与产品的市场化落地也呈加速趋势，数据表明截至 2021 年 8 月，中国已有 28 款不同的人工智能医疗产品获批三类医疗器械注册证[2]。

从全球来看，医疗人工智能的应用细分领域与中国略有不同。根据 Global Market Insight 的统计数据，药物研发在全球医疗人工智能市场中的占比最大，达到 35%。紧随其后的是医学影像人工智能，占比 25%，并将以超过 40% 的增速发展，预计 2024 年其规模将达到 25 亿美元。[3]

此外，基因组学分析是人工智能应用的又一重要领域。预计到 2022 年，该细分市场的规模仅在中国就将接近 300 亿元人民币[4]。基因测序与人工智能进一步结合，势必还会加速其发展，同时随之带来的测序时间缩短以及成本大幅降低，又会为医疗行业人工智能的应用创造更大的想象空间。

随着人工智能在更多医疗领域的运用，更多医疗数据也参与到各类机器学习和深度学习模型的训练中来，如何在提升模型性能的同时保证信息安全和隐私保护也是目前业界瞩目的焦点之一。因此，可信理念也在人工智能与医疗行业的结合中逐渐深入。

值得一提的是，利用人工智能方法来加速蛋白质结构预测也是目前广受关注的重要课题。以 AlphaFold2 为代表的新方案能够大幅加速蛋白质结构解析速度，揭示和呈现有机体内更多的信息秘密，是人工智能在生物学、医学和药学等领域落地的核心发力点之一。

同时，近年来广受关注的大语言模型（Large Language Model，LLM，以下简称“大模型”）技术在医疗领域的探索也获得了巨大的进展，并涌现出一批专门的医疗大模型。医疗大模型在学习性能、拟合效果、通用型和逻辑推理能力等方面的优势正帮助医疗行业加速智慧医疗进程的推进。

在中国，政策激励是加速医疗人工智能应用落地的关键因素之一。相关政府部门陆续推出了大量政策，从人才培养、技术创新、标准监管、行业融合、产品落地等多方位推动人工智能发展。其中，在 2018 年 1 月，国家标准化管理委员会指导下的《人工智能标准化白皮书（2018 版）》发布；同年 4 月，国务院印发《关于促进“互联网 + 医疗健康”发展的意见》，将推进“互联网 +”人工智能应用服务作为实施“健康中国”战略的重要举措，并表示将重点支持研发医疗健康相关的人工智能技术、医用机器人、大型医疗设备等。2021 年 10 月，由国家卫生健康委、国家中医药管理局印发的《公立医院高质量发展促进行动（2021-2025 年）》提出建设“三位一体”智慧医院。通过完善智慧医院分级评估顶层设计，鼓励有条件的公立医院加快应用智慧服务软硬件。2023 年 3 月由中共中央办公厅、国务院办公厅印发的《关于进一步完善医疗卫生服务体系的意见》指出发展“互联网 + 医疗健康”，加快推进互联网、人工智能、云计算等在医疗卫生领域中的应用，加强健康医疗大数据共享交换与保障体系建设。

医疗人工智能的应用趋势

人工智能在医疗健康领域的应用非常广泛，在从医学影像到健康管理、药物研发、慢性病管理以及生物学探索等诸多环节，都可发挥关键作用，并已在不同层级与不同细分领域的医疗机构呈现出各异的“职能”。其中，人工智能用于医学影像等场景主要服务于医院或其他医疗机构，其应用集中在疾病筛查方面。但囿于存在假阴性的情况，还需要医生审阅所有片子以防漏诊，致使此类应用在减轻医生工作量方面的效果并不显著。

[1] 数据援引自弗若斯特沙利文在2022年5月25日发布的报告《医疗智能行业白皮书》，https://www.frostchina.com/content/insight/detail?id=62f1ebd83a1cb46c9a9fd3ca
[2] 数据援引自中国信息通信研究院发布的《人工智能白皮书（2022年）》：http://www.caict.ac.cn/kxyj/qwfb/bps/202204/t20220412_399752.htm
[3] Global Market Insights report. 2018年4月. www.elecfans.com/rengongzhineng/592041.html
[4] 前瞻产业研究院。《2018-2023年中国基因测序行业市场前瞻与投资战略规划报告》。2018年。https://bg.qianzhan.com/trends/detail/506/180411-e7daa2c4.html

未来，人工智能在不同层级的医疗机构的应用方向可能会呈现出更加多元化的趋势，即在基层医院或第三方体检中心，其应用将以辅助筛查、辅助诊断以及慢性病管理为主；在三甲医院，则以提高医生工作效率为主；在健康管理方面，人工智能以支持单位和个人支付的健康体检为主要方向；在药物研发领域，人工智能应用又表现出不同特点，需要相关技术公司与大型药企、医药研究机构通力合作来推进。

虽然人工智能在医疗健康领域迅速得以应用，但源于数据、模型等方面的影响，目前仍然面临诸多挑战：

- **数据量。**模型越复杂，参数越多，所需要的训练样本量就越大。但是对许多复杂的临床场景而言，所需要的大量可靠数据却并不容易获得；
- **数据维度。**通常而言，数据维度越少，对真实世界的描述能力也越差，但高维数据处理面临着处理效率低、所需计算量大等问题；
- **数据质量。**一般而言，健康数据的组织化和标准化程度都不高，且数据分散、有噪声。在条件不好的诊所与基层医院，还存在电子病历信息缺失或有误、多机构间分散存储等问题，同时接口数据可靠性也很差；
- **数据孤岛现象。**作为关乎人们隐私信息的敏感领域，医疗数据泄露风险已经受到医疗机构的足够重视，但由此也催生出不同机构间数据相互隔离的数据孤岛现象。而单一医疗机构又难以聚集起足够的高质量训练数据，供 AI 模型训练学习所用；
- **模型的可解释性。**深度学习模型是个黑盒子，对如何得出结论没有明确的解释，其决策模式的权威性尚待验证；
- **模型的通用性。**首先是模型偏差，比如采用白种人患者数据进行训练的模型，可能在其他种族患者中效果不佳；还有就是模型互操作性差，即很难建立一个适用于两种不同电子病历系统的深度学习模型；
- **模型安全。**即便是训练有素的图像处理模型，也有可能因输入图像的扰动而受到不良影响，但这一扰动却无法被人察觉。此外，还存在数据“差之毫厘”就可能带来预测结果“失之千里”的问题。比如，轻微改变患者电子病历数据中的实验检测值，就可能极大影响模型对住院死亡率的预测。

针对这些挑战，医疗和人工智能等领域的专家已经提出多项应对措施，来优化应用环境，提高应用实效：

- **收集大规模和多样化的健康数据。**广泛收集来自不同种族、民族、语言和社会经济地位患者的数据，并对其进行标准化和集成；
- **提高数据质量。**从提供可靠、高质量的数据输入入手，继之再利用工具提高数据收集的质量，如进行错误纠正、发出关于缺失数据的警告等；
- **融入临床工作流程。**将深度学习融入现有电子病历系统的管理，提高临床医生的工作效率和数据采集的实时性；
- **构建高维学习模型。**引入百万级乃至上亿级的规则，通过高维学习模型，大幅提升预测和识别的准确率；
- **推进医疗大模型建设。**借助拥有海量参数的大模型，以更优异的学习效果来实现更精准、更可靠的医疗智能化水平。
- **法制化规范化。**针对诸如计算机黑客篡改数据，从而影响深度学习模型的结果等信息安全问题，要制定相应法规，保护分析模型。

同时，为推动多源医疗数据进行更安全的交互、传输和聚合，解决因数据孤岛所造成的高质量训练数据不足问题，各方专家正积极探索引入联邦学习方法等安全性更高的数据协作方式和更完善的 AI 模型训练架构，以便在降低隐私泄露风险的前提下，以更多高质量数据构建起安全可信的多源数据协同方案，提升医疗 AI 应用效能，使 AI 技术更高效、安全地服务于医疗健康。

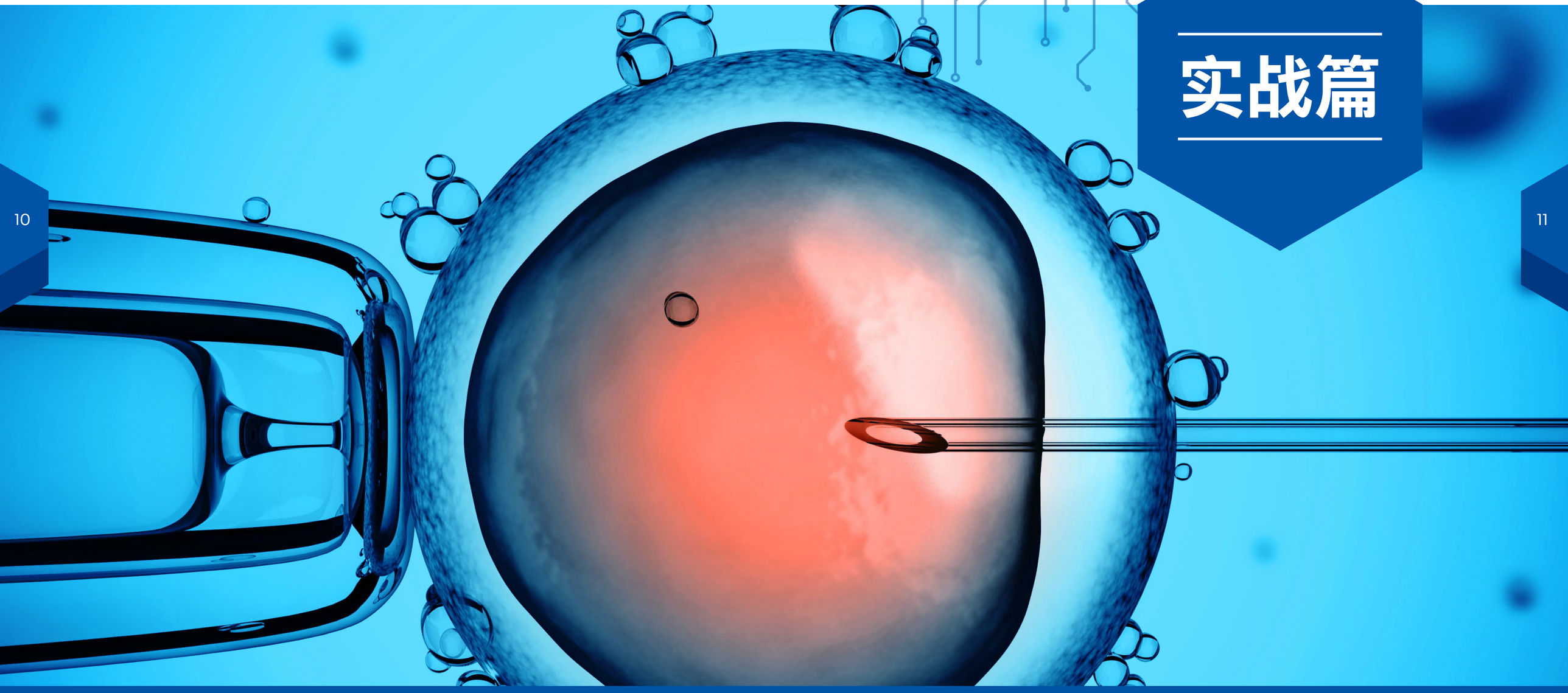
人工智能在医疗健康领域的应用场景

医疗健康是人工智能应用落地最具潜力的领域之一，对此业界已有共识。伴随着应用的不断深入，人工智能将在以下多种医疗健康应用场景中大显身手：

- **慢病管理与疾病监测。**基于患者体征对（潜在）慢性疾病进行风险预估，从而通过早期干预，大大降低患者的医疗费用；
- **临床预测分析。**例如，基于电子病历数据评估在院内感染疾病的风险，根据运营模型预测患者再入院率，根据财务模型制定捆绑销售服务方案等；
- **慢性病管理。**利用数据采集方法（例如物联网），构建基于人工智能方法的慢性病评估及筛查模式，提升慢性病的预测和早期诊断能力；
- **病历搜索与质量控制。**精准提取医疗文本中的关键信息，进行医学实体识别，进而实现灵活的全量电子病历搜索；
- **虚拟现实助手。**通过虚拟现实会话，参与到患教活动中，帮助患者清楚了解其病因，使医患沟通更有效；
- **智能导诊。**通过语音、触屏等多种交互方式，更好地提供院内导航、导诊、导医，提升精准分诊、健康咨询、健康宣教等服务的水平；
- **影像辅助诊断。**帮助放射科医生快速筛除正常影像，提高医生的病例处理效率；提高分析影像的准确度，缩短诊断结果报告时间，提升医疗系统的诊断能力；
- **病理分析。**例如，高效、准确地对送检物进行检测和分类；
- **基因组学分析。**用以大幅降低基因测序成本，快速精确实现规模庞大的基因组数据分析，为疾病的诊断和治疗等提供支持；
- **蛋白质结构预测。**通过深度学习方法，加速蛋白质结构的解析，为生物学、医学、药学乃至农业、畜牧业等领域的未来研究与发展提供高质量的生物学假设；
- **药物发现。**加快药物研发效率，降低成本。

在下一章 “实战篇” 中，我们将结合英特尔与东软、西门子、盈谷、第四范式、汇医慧影、致远慧图、卫宁健康、医渡云、锘崴科技、江丰生物以及惠每科技等产业以及江丰生物等产业伙伴与客户在医疗人工智能领域的实战案例，详细介绍项目的背景、实施过程，以及取得的经验与成果，还将结合各应用场景提供相对应的软、硬件配置推荐。

实战篇



OpenVINO™ 提升医疗图像AI推理效率

医学影像处理中的图像分割

传统医学影像图像分割方法

计算机视觉中的图像分割[5]是指以图像中的自然边界，例如物体轮廓、线条等，将图像切分为多个区域，其目的是用于简化或改变图像的表现形式，使之更易解读和分析。在计算机方法中，这一过程通常会被解构为将图像中的每个像素加上标签，使具有相同标签的像素有着某种共同视觉特性，例如颜色、亮度、纹理等，由此进行的度量或计算得出的一定区域的像素特性相似，而邻接区域则有着很大的不同。

作为计算机视觉技术的重要分支，图像分割已在医学影像处理、工业机器人、智能交通、指纹识别以及卫星图像定位等多个行业和领域获得广泛应用。在医学影像处理领域，图像分割已在诸多病理位置定位、组织体积测量、解剖学研究、计算机辅助手术、治疗方案制定以及临床辅助诊断等多个细分领域证明了其价值。

传统的图像分割方法主要有以下几种常见方法：

- **基于聚类的方法**：聚类法是基于K-均值算法，将图像迭代分割成K个聚类。该算法中，分割图像中像素与聚类中心之间都有着相似的距离偏差，距离偏差通常采用颜色、亮度、纹理、位置等指标。该算法具有良好的收敛性；
- **基于阈值的方法**：该方法是通过计算图像的一个或多个灰度阈值后，将每个像素的灰度值与阈值相比较，最后进行归类的方法；
- **基于边缘的方法**：该方法是根据图像中自然边缘的灰度、颜色、纹理等特性的突变性来对图像进行分割。一般来说，基于边缘的分割方法依赖于灰度值边缘检测，当边缘灰度值呈现阶跃型等变化时，判断为图像边缘；
- **基于区域的方法**：该方法是根据图像的相似性来对图像进行分割，其判断原则是根据相邻像素点的灰度、颜色、纹理等特性是否存在相似性，如有相似，则扩大像素点的集合。

基于深度学习的图像分割方法

随着近年来AI技术的飞速发展，尤其是在图像领域，基于AI技术的图像识别、图像处理应用已经被用在很多场景中，其对各类医学影像的分析识别能力已经超过人类。与卷积神经网络（Convolutional Neural Network, CNN）类似的模型，是目前基于AI的图像分割技术中常见的网络模型。这其中，全卷积网络（Fully Convolutional Network, FCN）、U-Net和V-Net是常见的几种基于深度学习的图像分割方法。

■ FCN

CNN 的典型用途是对任务进行分类。对图像处理而言，它的输出是单个类别标签。在生物医学的图像分割处理中，期望的输出应该包括定位，即应该将类标签分配给每个像素。作为卷积神经网络的升级扩展版本，如图 2-1-1 所示，FCN[6] 遵循编码、解码的网络结构模式级联了卷积层和池化层。卷积层和最大池化层有效降低了原始图像的空间维度。同时，FCN 使用 AlexNet 作为网络的编码器，采用多重转置卷积重复扩展的方式，对编码器最后一个卷积层输出的特征图进行上采样，直到特征图恢复到输入图像的分辨率，因而，可以实现像素级别的图像分割。

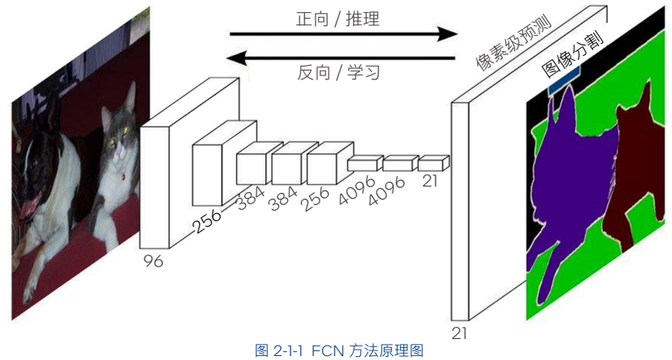


图 2-1-1 FCN 方法原理图

[5]关于图像分割的描述，部分参考：Linda G. Shapiro and George C. Stockman (2001): “Computer Vision”, pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[6]关于FCN的相关技术描述，摘选自UC Berkeley jonlong、shelhamer和trevor的论文《Fully Convolutional Networks for Semantic Segmentation》：https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

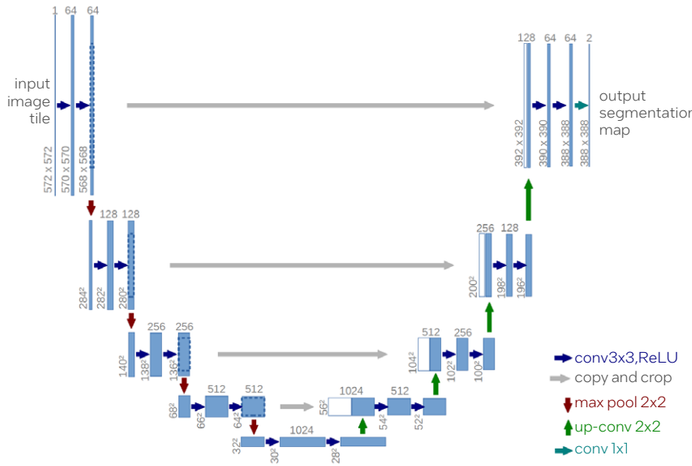


图 2-1-2 U-Net 拓扑

■ U-Net

作为 FCN 网络的一个改进版本，U-Net 具有一个鲜明的 U 型结构，其拓扑图如 2-1-2 所示，其在每个 Encoder 上都会进行 4 次上采样，这使得分割图恢复边缘等信息会更为精细。同时，在同一个 stage 上，U-Net 都采用了跳跃连接（skip connection），而不是直接在高级语义特征上进行监督和 loss 反传，这样就可以保证最后得到的特征图融合了多的低层级（low-level）特征，也使得不同尺度的特征得到了融合，从而可以进行多尺度预测（Multi-Scale Prediction）和深度监督（Deep Supervision）。另外，U-Net 在网络后部补充了一个与前面类似的网络，形成 U 性结构。其中池化运算符由上采样运算符替换，因此增加了输出的分辨率。同时，为了定位，模型从收缩路径的高分辨率特征与上采样输出相结合。连续卷积层可以采用 relu 激活函数来对原始图片进行降采样操作，从而获得更精确的输出。

医学影像在实际应用中也有其独有的特性。我们可以看到，一般胸片影像是胸片 CT，而眼底检查则是眼底 OCT，均为针对一个指定器官的成像，而非全身。而器官本身结构比较固定，语义信息并非特别丰富。所以高级语义信息和低层级特征就显得非常重要，而 U-Net 的 U 型结构和跳跃连接在这种场景中，可以发挥出更大作用。近年来，U-Net 在医学影像分割领域良好的应用效果，已在很多部署中得到充分了证明。

■ V-Net

V-Net 可以视为 3D 版本的 U-Net，如图 2-1-3 所示，它与 U-Net 有着类似的拓扑形态，适用于三维结构的医学影像分割。V-Net 能够实现基于 3D 图像的端到端图像语义分割，并通过类似于残差学习的 trick 来对网络进行改进。

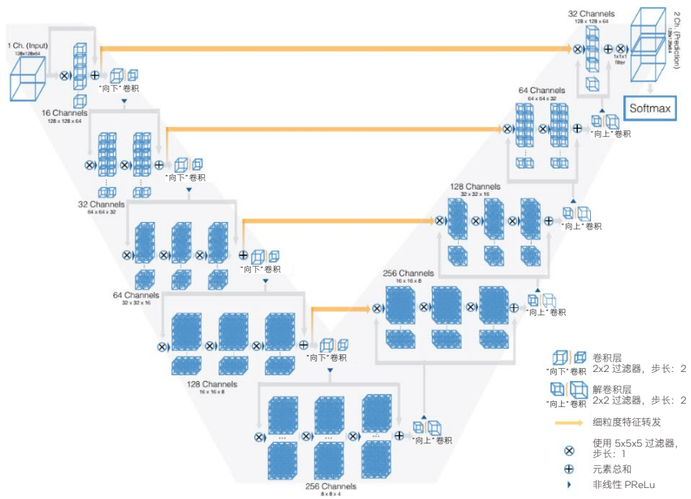


图 2-1-3 V-Net 拓扑思想

软硬件配置建议

对于在医疗行业中构建基于深度学习的图像分割方法，可以参考以下基于英特尔® 架构平台的软硬件配置来完成。

名称	规格
处理器	英特尔® 至强® 金牌 6240 处理器或更高
超线程	ON
睿频加速	ON
内存	16GB DDR4 2666MHz* 12及以上
存储	英特尔® 固态盘 D5 P4320系列及以上
操作系统	CentOS Linux 7.6 或最新版本
Linux 核心	3.10.0 或最新版本
编译器	GCC 4.8.5 或最新版本
Python 版本	Python 3.6 或最新版本
TensorFlow 版本	R1.13.1 或最新版本
OpenVINO™ 工具套件	2019 R1 或最新版本
Keras 版本	2.1.3 或最新版本

U-Net分割网络的优化方法

基于英特尔® 架构的优化方法

将传统的CNN图像分割方法用于医学图像时，往往存在以下困难：

- CNN通常都是应用于分类，生物医学图像则更关注分割以及定位的任务；
- CNN需要获取大量的训练数据，而医学图像很难获得相应较大规模的数据。

以往在应对上述困难时，通常采用滑窗的方法，即为每一个待分类的像素点取周围的一部分邻域输入。这种方法好处有两点：首先，这一方法能够在滑窗的同时完成定位工作；其次，每次动作都会取一个像素点周围的邻域，可以大大增加训练的数据量。但是，这一方法也有两个缺点：一是通过滑窗所取的块之间有较大的重叠，会导致训练和推理速度变慢；二是网络需要在局部准确性和获取上下文之间进行取舍，因为如果滑窗取的块过大，就需要更多的池化层，定位准确率会降低，而取的块过小，则网络只能看到很小的一部分上下文。

基于英特尔® 架构平台开展的一系列优化，可以从另一个层面帮助用户解决以上问题。这些优化方法包括：调整处理器核心数量、引入非统一内存访问架构（Non- Uniform Memory Access Architecture，NUMA）技术，以及面向深度神经网络的英特尔® 数学核心函数库（Intel® Math Kernel Library for Deep Neural Networks, 英特尔® MKL-DNN），从而为U-Net图像分割法提供多层次的优化。优化步骤如下：

■ 环境变量设置

首先，需要对环境变量进行设置，如以下所示，命令包括：清空系统的缓存（cache），将处理器设置为性能优先的模式，即运行在最高频率，打开处理器的睿频加速。

```
1. echo 1 > /proc/sys/vm/compact_memory
2. echo 3 > /proc/sys/vm/drop_caches
3. echo 100 > /sys/devices/system/cpu/intel_pstate/min_perf_pct
4. echo 0 > /sys/devices/system/cpu/intel_pstate/no_turbo
5. echo 0 > /proc/sys/kernel/numa_balancing
6. cpupower frequency –set –g performance
7.
8. export KMP_BLOCKTIME=1
9. export KMP_AFFINITY=granularity=fine,verbose,compact,1,0
10. export KMP_OMP_NUM_THREADS=20
```

- KMP_BLOCKTIME 设置为 1，是设置某个线程在执行完当前任务并进入休眠之前需要等待的时间，通常设置为 1 毫秒；
- KMP_AFFINITY 设置为 Compact，是表示在该模式下，线程绑定按计算核心的计算要求优先，先绑定同一个核心，再依次绑定同一个处理器上的下一个核心。此种绑定适用于线程之间具有数据交换或有公共数据的计算情况，优势在于可以充分利用多级缓存的特性；
- OMP_NUM_THREADS 设置为 20，是将并行执行线程的数量设定为所需的物理核心数。

■ 测试代码中添加线程控制

```
1. config = tf.ConfigProto()
2. config.allow_soft_placement = True
3. config.intra_op_parallelism_threads = FLAGS.num_intra_threads
4. config.inter_op_parallelism_threads = FLAGS.num_inter_threads
```

如上述设置命令所示，在进行 tf.ConfigProto() 初始化时，我们也可以通过设置 intra_op_parallelism_threads 参数和 inter_op_parallelism_threads 参数，来控制每个操作符 op 并行计算的线程个数。二者的区别在于：

- intra_op_parallelism_threads 控制运算符 op 内部的并行，当运算符 op 为单一运算符，并且内部可以实现并行时，如矩阵乘法、reduce_sum 之类的操作，可以通过设置 intra_op_parallelism_threads 参数来并行，intra 代表内部。

- inter_op_parallelism_threads 控制多个运算符 op 之间的并行计算，当有多个运算符 op，并且它们之间比较独立，运算符和运算符之间没有直接的路径 Path 相连时，TensorFlow 会尝试并行地对其进行计算，并使用由 inter_op_parallelism_threads 参数来控制数量的一个线程池。

通常而言，intra_op_parallelism_threads 设置为单个处理器的物理核心数量，而 inter_op_parallelism_threads 则设置为 1 或者 2。

■ 利用 NUMA 特征来控制处理器计算资源的使用

数据中心使用的服务器，通常都是配置两颗或更多的处理器，多数都采用 NUMA 技术，使众多服务器像单一系统那样运转。处理器访问它自己的本地存储器的速度比非本地存储器更快一些。为了在这样的系统上获取最好的计算性能，需要通过一些特定指令来加以控制。Numactl 就是用于控制进程与共享存储的一种技术机制，也是在 Linux 系统中广泛使用的计算资源控制方法。具体使用方法如下所示：

图 2-1-4 用 NUMA 特征来控制处理器计算资源的使用

```
1. numactl –C 0-19,40-59 –m 0 python3 test.py
```

上述指令表示的是 test.py 在执行的时候只使用了处理器 #CPU0 中的 0-19 和 40-59 核，以及处理器 #CPU0 对应的近端内存。

■ 采用面向英特尔® MKL-DNN 优化的 TensorFlow

为了使用户在通用处理器平台上进行高效的 AI 计算，英特尔针对众多主流的深度学习开源框架进行了大量的优化，包括目前在工业界和学术界使用十分广泛的 TensorFlow。

通过使用英特尔® MKL-DNN 优化的多种原语（Primitive），英特尔对 TensorFlow 进行了优化。英特尔® MKL-DNN 是从 TensorFlow 1.2 开始添加的。除了在训练基于 CNN 的模型时能显著提升性能之外，使用英特尔® MKL-DNN 进行编译还可以创建针对英特尔® 高级矢量扩展指令集（Intel® Advanced Vector Extensions，英特尔® AVX）、英特尔® AVX 2 和英特尔® AVX-512 进行优化的二进制文件，从而得到一个经过优化且与大多数现代（2011 年后）处理器兼容的文件。

参考文献：

- https://www.tensorflow.org/guide/performance/overview?hl=zh_cn
- https://software.intel.com/zh-cn/articles/tensorflow-optimizations-on-modern-intel-architecture

* 更多英特尔® MKL-DNN 的技术细节，请参阅本手册技术篇相关介绍。

U-Net基于英特尔® 架构优化后的测试及结果

通过以上四个方面的优化，U-Net在基于英特尔® 架构的处理器平台上的性能得到了显著提升，测试结果如下图所示[7]：

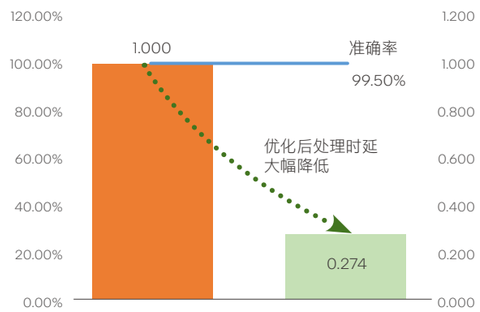


图 2-1-5 基于英特尔 ® 架构优化前后性能对比

基于 OpenVINO™ 工具套件英特尔® 发行版对 U-Net 进一步优化

为满足客户在实际应用场景中的需求，在上述结果的基础上，英特尔又基于 OpenVINO™ 工具套件英特尔® 发行版（以下简称 “OpenVINO™ 工具套件”）对 U-Net 图像切割方法实施了进一步的优化，具体优化步骤如下：

■ 模型转换

由于原有的模型是基于 Keras 进行训练，生成的模型为 hdf5 格式，这种格式的模型无法直接作为 OpenVINO™ 工具套件的输入，需要先进行格式转换，操作命令如下：

```
1. git clone https://github.com/amir-abdi/keras_to_tensorflow.git
2. cd keras_to_tensorflow
3. python3 keras_to_tensorflow.py --input_model=../unet/unet_membrane.hdf5 --
   output_model=../unet/full_unet.pb ##做模型转换
```

[7] 测试配置为：处理器：双路英特尔® 至强® 金牌 6148 处理器，2.40GHz；核心 / 线程：20/40；内存：16GB DDR4 2666MHz * 12；硬盘：英特尔® 固态盘 SC2BB480G7；BIOS：SE5C620.86B.02.01.0008.031920191559；操作系统：CentOS Linux 7.6；Linux 内核：3.10.0-957.21.3.el7.x86_64；gcc 版本：7.2；Python 版本：Python 3.6；TensorFlow 版本：R1.13.1。

■ 将模型通过 OpenVINO™ 工具套件的 mo.py 转换成 xml 文件和 bin 文件

命令如下：

```
1. python3 /opt/intel/openvino/deployment_tools/model_optimizer/mo.py --framework tf --
   input_model full_unet.pb --data_type FP32 --output_dir ./ --input_shape [28,512,512,1]
```

■ 通过Inference Engine 来进行模型推理

命令如下：

```
1. python3 segmentation_demo.py -m /home/worker/unet/full_unet.xml -
   i /home/worker/0.png -l /home/worker/openvino/intel64/Release/lib/libcpu_extension.so
```

其中，做推理的代码包含如下逻辑模块：

```
1.  #Load input data ##数据数据预处理及导入，包括数据格式统一（医学图像 dcm 格式转化为
    jpg）；执行图像缩放，多通道扩增，归一化处理等操作；
2.  #Loading model to the plugin net = IENetwork.from_ir(model=model_xml, weights=model_bi
    n) ##其中 xml 文件为网络结构，bin 文件为权重参数）
3.  input_blob = next(iter(net.inputs))##确定模型的输入
4.  out_blob = next(iter(net.outputs)) ##确定模型的输出
5.  exec_net = plugin.load(network=net) ##模型导入
6.  # Start sync inference
7.  res = exec_net.infer(inputs={input_blob: images}) ##数据推理过程
8.  # Processing output blob
9.  res = res[out_blob] ##提取推理结果
10. #Visualization result
```

基于 OpenVINO™ 工具套件的优化结果

优化结果如图 2-1-6 所示，最左列为 CT 原图，中间列是未优化时的图像分割结果，最右列是通过 OpenVINO™ 工具套件优化之后生成的图像分割结果。可以看出，通过 OpenVINO™ 工具套件优化后生成的图像分割结果，在准确率上与未优化时基本保持一致，但在推理速度上却远高于未优化时[8]。

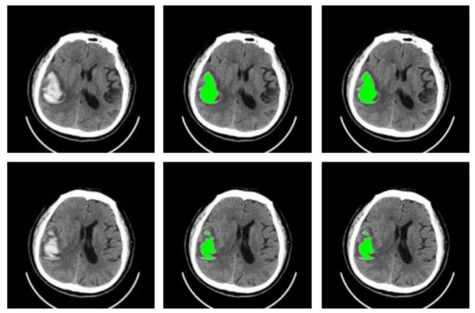


图 2-1-6 基于 OpenVINO™ 工具套件对 U-Net 的优化结果

* 更多 OpenVINO™ 工具套件的技术细节，请参阅本手册技术篇相关介绍。

Dense U-Net 图像分割方法

英特尔® 深度学习加速（Intel® Deep Learning Boost, 英特尔® DL Boost）技术

英特尔® 至强® 可扩展处理器从第二代起，不仅以优化的微架构、更多的内核及更快的内存通道带来了计算性能的提升，更面向 AI 应用提供了更为全面的加速能力，在其集成的英特尔® 深度学习加速技术（VNNI 指令集）中，加入了对 INT8 的支持，为用户提供了高效的 INT8 深度学习推理加速能力，这一能力将有效提升 U-Net 图像分割方法的执行效率。

英特尔® 深度学习加速技术通过 VNNI 指令集来支持 8 位或 16 位低精度数值相乘，这对于需要执行大量矩阵乘法的深度学习计算而言尤为重要。它的导入使得用户在执行 INT8 推理时，对系统内存的要求最大可减少 75%[9]，而对内存和所需带宽的减少，也加快了低数值精度运算的速度，从而使系统整体性能获得大幅提升。

与以往的 FP32 模型相比，INT8 模型具有更小的数值精度和动态范围，因此在图像切割等深度学习中采用 INT8 推理方式，需要着重解决计算执行时的信息损失问题。一般地来讲，INT8 推理功能可以通过量化校准的方式来形成待推理的 INT8 模型，进而实现将 FP32 在信息损失最小化的前提下转换为 INT8 的目标。

以图像分析应用为例，从高精度数值向低精度数据转换，实际是一个边计算边缩减的过程。换言之，如何确认缩减的范围是实现信息损失最小化的关键。在 FP32 向 INT8 映射的过程中，采用根据数据集校准的方式，来确定映射缩减的参数。在确定参数后，平台再根据所支持的 INT8 操作列表，对图形进行分析并执行量化 / 反量化等操作。量化操作用于 FP32 向 S8（有符号 INT8）或 U8（无符号 INT8）的量化，反量化操作则执行反向操作。

基于 OpenVINO™ 工具套件进行 FP32 模型到 INT8 模型的转换

通常地，通过神经网络训练好的模型是单精浮点精度的，即 FP32，用户可以将这样的模型直接部署在实际应用场景中，并通过量化技术得到低精度模型，比如 INT8 模型在保证模型

[8] 相关验证测试配置为：处理器：双路英特尔® 至强® 金牌 6148 处理器，2.40GHz；核心/线程：20/40；内存：16GB DDR4 2666MHz* 12；硬盘：英特尔® 固态盘SC2BB480G7；BIOS：SE5C620.86B.02.01.0008.031920191559；操作系统：CentOS Linux 7.6；Linux内核：3.10.0-957.21.3.el7.x86_64；gcc版本：4.8.5；Python版本：Python 3.6；OpenVINO™ 工具套件：2019 R1；Keras：2.1.3。

[9] 数据源引自https://software.intel.com/en-us/articles/lower-numerical-precision-deep-learning-inference-and-training

精度的基础之上可以提供效率更高的模型推理应用，通常情况下模型精度的损失小于 1%。

OpenVINO™ 工具套件从 2018 R4 版本开始提供 FP32 模型到 INT8 模型的转换功能，并且从 2019 R1 版本开始，支持基于第二代英特尔® 至强® 可扩展处理器所集成的英特尔® 深度学习加速技术。

OpenVINO™ 工具套件中的模型优化器基本工作和部署流程为：首先工具套件会将训练好的、基于开放神经网络交换（Open Neural Network Exchange，ONNX）训练的模型进行转换和优化，生成 FP32 格式的 xml 文件和 bin 文件，其中的优化包含节点融合、批量归一化的去除和常量折叠等方法；然后，通过 OpenVINO™ 工具套件中的转换工具将 FP32 格式的文件转换为 INT8 格式的 xml 文件和 bin 文件，在转换的过程中需要用到一个小批量的验证数据集，并且会将转换量化过程中的统计数据存储下来，以便在后续的推理时确保精度不受影响。上述的转换流程是离线运行的，也就是只要转换一次即可，详细做法如图 2-1-7 所示：

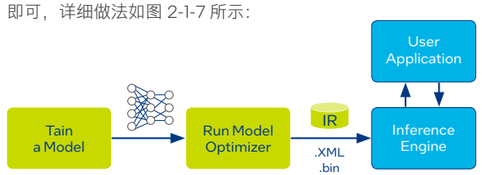


图 2-1-7 基于 OpenVINO™ 工具套件的 FP32 模型到 INT8 模型的转换[10]

按照上述模型转换之后，得到初步模型，其性能如下图所示：

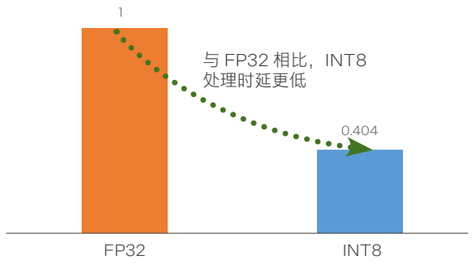


图 2-1-8 FP32 与 INT8 的时延性能对比

通过对两种模型进行性能分析可以看出，FP32 模型中的重排序操作（Reorder Ops）占据了大量的执行时间，在 INT8 模型中，重采样（Resample Ops）只支持 FP32 的操作，连接操作（Concat Ops）执行时间过长，而本来占比最高的卷积操作（Convolution Ops）在整个模型运行中占据的时间比例反而少。因此，需对其进行进一步的优化。

如图2-1-9所示，经过优化，模型的延迟有了大幅度的降低。

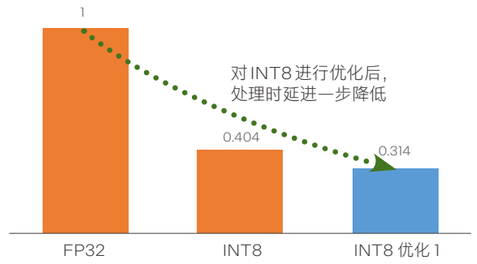


图 2-1-9 优化后的 INT8 模型时延性能对比

此时再将 INT8 模型进行逐层分析，可以看到相比之前已经有了很明显的提升。但在优化之后的模型中，Concat Ops 所占据的执行时间还是较长。为了进一步提升模型的吞吐量，需对 Concat Ops 进行特定优化，并且不再使用英特尔® MKL-DNN 中的原语，而是要进行定制化，详细代码如下所示：

```
1.  for (size_t i = 0; i < num_src; i++) {
2.      const MKLDNNMemory& src_mem = getParentEdgeAt(i)->getMemory();
3.      channels.push_back(src_num.GetDims()[1]);
4.      src_ptrs.push_back(reinterpret_cast<const uint8_t*>(src_mem.GetData()));
5.      dst_ptrs.push_back(dst_ptr + channels_size);
6.      channels_size += src_mem.GetDims()[1];
7.  }
8.
9.  parallel_for(iter_count, [&](int i){
10.     for (int j = 0; j < src_ptrs.size(); j++) {
11.         memcpy(dst_ptrs[j] + (i * channels_size), src_ptrs[j] + i * channels[j], channels[j]);
12.     }
13. });
```

上述优化主要的目的是，实现并行化地批量拷贝数据到指定位置。通过此类型的优化，模型性能有了进一步的提升。此时的模型执行时间基本达到了理想状况，最终优化结果如图 2-1-10 所示：

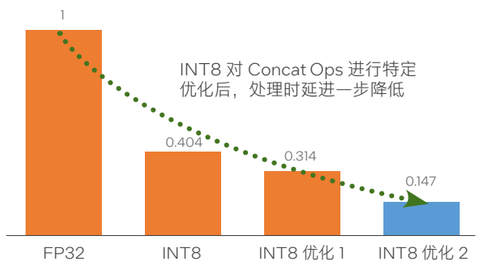


图 2-1-10 进一步优化后的 INT8 模型时延性能对比

从性能分析可以获知，此时模型运行占比最高的原语成了卷积操作，完全符合本实例中 Dense U-Net 模型本应有的效果。

[10] 本图来源于 https://docs.openvinotoolkit.org/latest/_docs_MO_DG_Deep_Learning_Model_Optimizer_DevGuide.html

应用案例

东软 eStroke 影像平台

■ 背景

医疗影像的快速和准确判读对医院医疗技术能力有着一定的要求，同样也需要专业的影像医生进行判读，十分依赖医生的判读水平。为应对这一挑战，医疗行业需要一种即便在基层医院医生判断水平不足的情况下，仍然可以快速准确地对相关医学影像进行分析的工具。现在，基于深度学习的医学影像判读已经逐步走入医疗机构，帮助应对以上问题。

■ 方案与成效

eStroke 影像平台具有以下优势：

- **支持多模态影像学设备。**其中包括电子计算机断层扫描（Computed Tomography，CT）、核磁共振成像（Magnetic Resonance Imaging，MRI）图像等 16 排以上多层螺旋 CT 以及 1.5T 以上 MRI；
- **实现全流程自动化。**从医院设备扫描序列开始到影像后处理分析，一直到输出影像诊断报告，均无需人工干预；
- **能够接入互联网医疗诊治技术应用研究平台等外部诊疗系统。**支撑开展远程急救、移动急救、高危人群智能预警及干预、疾病联合救治、虚拟手术等技术研发和工程化。

以 eStroke 影像平台为载体，东软与英特尔携手，基于 U-Net 模型对平台中的医学影像进行图像分割处理，根据 eStroke 平台对灌注成像的各个参数，包括 CBF、CBV、MTT 和 TMAX 的计算，并结合以上参数通过左右脑循环的对称性，如图 2-1-11 所示，进一步推理出用于医学诊断的病灶所在区域。

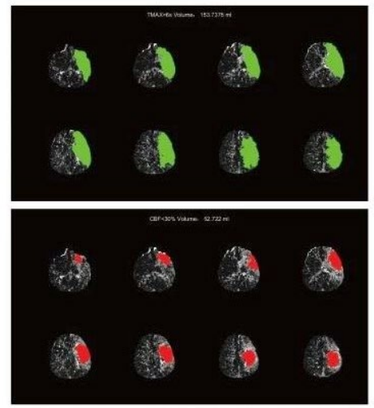


图 2-1-11 通过 TMAX＆CBF 异常区域计算出的病灶区域

该方案基于面向英特尔®架构优化的 TensorFlow（基于英特尔® MKL-DNN 优化）以及 OpenVINO™ 工具套件进行了优化，使基于 U-Net 模型的深度学习推理在保证准确性的同时，推理时间得以大幅减少。这对于争分夺秒的脑卒中诊治而言，无疑有着重大的实践意义。如图 2-1-12 所示，在推理准确性基本一致的情况下，采用两个工具优化后的方案与未经优化的方案对比，推理延迟分别降低 72.6% 和 85.4%[11]。

图 2-1-12 东软 U-Net 图像分割各方案性能对比

[11] 该数据所使用的测试配置为：处理器：双路英特尔® 至强® 金牌 6148 处理器，2.40GHz；核心/线程：20/40；内存：16GB DDR4 2666MHz * 12；硬盘：英特尔® 固态盘SC2BB480G7；BIOS：SE5C620.86B.02.01.0008.031920191559；操作系统：CentOS Linux 7.6；Linux内核：3.10.0-957.21.3.el7.x86_64；gcc版本：7.2（TensorFlow）& 4.8.5（OpenVINO）；Python版本：Python 3.6；TensorFlow版本：R1.13.1；OpenVINO™ 工具套件：2019 R1；Keras：2.1.3。

西门子医疗利用英特尔® 深度学习加速技术，推进诊疗中的 AI 应用

■ 背景与挑战

过去，医生往往需要凭借经验来对 MRI 影像进行判读，不仅费时费力，且错误率较高，在解释图像时也容易受到主观因素的影响，导致漏诊和误诊。

随着 AI 技术的发展，西门子医疗正在开展一系列创新医疗 AI 应用研究，并将成果纳入实际应用。但要将这些 AI 能力真正应用到医疗实践中，还面临着一系列的挑战。

首先，AI 应用对临床诊疗带来延迟。AI 应用需要与各类检查仪器生成的数据保持同步，并保证 AI 推理具备高吞吐、低延迟的特性，才能让基于 AI 的医疗系统服务更多患者。其次，AI 应用应当尽可能与临床诊疗流程进行融合，以便节省时间，并提高测量和诊断之间的一致性和准确性。

为此，西门子医疗与英特尔一起，基于通用处理器平台来开展针对 MRI 影像的判读和测量，实施高效的 AI 推理工作。双方不仅利用深度学习的方法对来自 MRI 的影像进行了 AI 判读研究，同时基于第二代英特尔® 至强® 可扩展处理器以及 OpenVINO™ 工具套件等，进行了优化工作，使推理速度大幅提升，为临床医学诊疗提供了强有力的支撑。

■ 方案简介及实施效果

在本案例中，西门子医疗与英特尔一起合作，优化了基于第二代英特尔® 至强® 可扩展处理器构建的医疗检测和量化模型。该 AI 模型基于 Dense U-Net，可对检测对象进行语义分割。AI 模型的输入是检测对象的 MRI 图像的堆叠，输出则是检测对象的不同区域以及结构，其中每个结构都会被颜色编码。这样可以将原先需要人工识别标注的过程智能化，从而加快影像判读速度，其整体工作流程见图 2-1-13 所示。

第二代英特尔® 至强® 可扩展处理器为该 AI 模型的推理提供了高效、灵活和可扩展的平台，特别是经与 OpenVINO™ 工具套件的紧密结合，有效地加速了针对视觉应用的深度学习推理，提高了诊疗过程中至关重要的诊断与决策的速度和准确性。同时，处理器集成的英特尔® 深度学习加速技术，具有全新的矢量神经网络指令（VNNI），能够进一步加速深度学习中的各种计算密集型操作，让图像分类、图像分割、目标检测等 AI 应用在英特尔® 处理器平台上推理效率变得更高。英特尔® 深度学习加速技术对 INT8 良好的支持能力，使其可以将 FP32 训练模型转化为 INT8，在保持准确性的同时大幅提升推理速度。

在本案例中，深度神经网络（例如 Dense U-Net）经过训练后被用以识别检测对象区域，神经网络的权值通常采用浮点数值（FP32）来表示，因此模型通常情况下会通过 FP32 精度来进行训练和推理。但 INT8 同样可以在损失很小的准确率（通常 <0.5%，本案例中可达到 <0.001%）情况下来提升推理速度[12]。

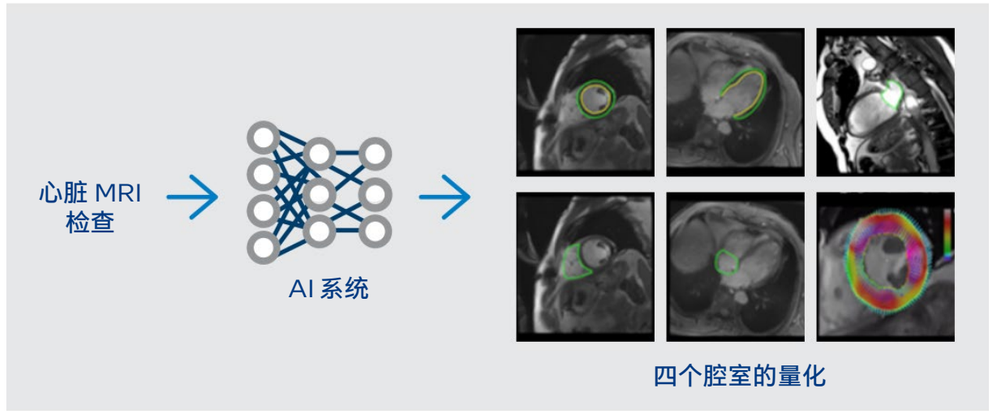


图 2-1-13 西门子医疗与英特尔一起构建面向 MRI 的 AI 分析能力

[12] 该数据援引自 Journal of the American College of Cardiology, 2017.

通过英特尔® 深度学习加速技术和 OpenVINO™ 工具套件提供的 FP32 到 INT8 的转换工具，英特尔帮助西门子医疗实现了在保持准确率的情况下，以更高的速度来进行推理运算的能力。图 2-1-14 显示了使用 INT8 模型前后输出的图像，可以直观地看到，两者的精度基本一致。

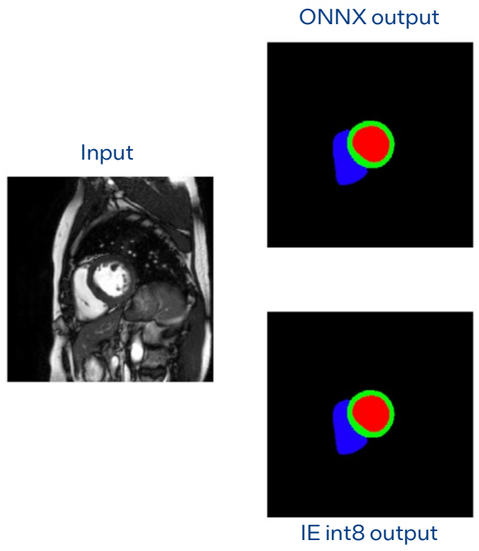


图 2-1-14 使用 INT8 模型前后的输出精度对比

从推理速度来看，该方案在基于第二代英特尔® 至强® 可扩展处理器、英特尔® 深度学习加速技术以及 OpenVINO™ 工具套件进行优化后，面向 MRI 的 AI 分析能力得以大幅增强。一方面，MRI 影像的处理速度获得了显著增强，达到了 200 FPS（帧每秒），为相关 MRI 在临床上的近实时应用开辟了可能；另一方面，优化后的解决方案，在量化和执行模型时，在几乎没有降低精度的情况下，性能可以提升到未优化方案的 5.5 倍[13]。

GE 医疗利用英特尔技术与产品，优化深度学习模型，提升 CT 图像推理性能

■ 背景

CT 检查是现代医学中最常用的检查手段之一。其通过 X 射线束对人体层面进行扫描，并得到相关部位的断面或立体图像，从而发现人体的病变情况。CT 检查虽然有着极为重要的临床意义，但 CT 切片图像的检查在传统上往往依赖经验丰富的医生进行人工读片，不仅效率较低，且受医生主观性的影响也会带来误诊、漏诊。

现在，通用电气医疗集团（以下简称 “GE 医疗”）正利用深度学习的方法，对 CT 切片图像进行分类和标记，这更便于医生寻找到微小病灶，并将其用于研究或临床比较。在 2018 年的医学成像光学会议（SPIE）上，GE 医疗发表了一篇关于基于 AI 的结构分类器的论文，其 CT 成像专家使用 Python 语言、TensorFlow 框架以及 Keras 库构建并训练了新的 AI 模型。通过与英特尔开展的深入技术合作，双方正利用英特尔® 至强® 处理器、英特尔® 深度学习部署工具（Intel® Deep Learning Deployment Toolkit, 英特尔® DLDT）等产品与技术，来优化其面向 CT 推理的解决方案。

■ 方案与成效

方案中引入了英特尔® DLDT 来优化深度学习模型，并在英特尔® 至强® 平台上展现出更好的推理性能。

英特尔® DLDT 是 OpenVINO™ 工具套件中，专门用于深度学习模型的推理加速部件。通过该工具，训练收敛的模型可以在多种英特尔® 平台上获得更高的数据处理能力，以及更低的数据处理延时。其可以对多种主流深度学习开源框架训练好的模型进行转换和优化，生成独立于深度学习框架的 bin 文件和 xml 文件。其中 bin 文件用于存放深度学习模型的权重，以二进制形式存储，而 xml 文件则描述深度学习模型的网络结构，二者结合起来共同解析模型。这使得模型的表征文件不依赖于任何深度学习框架，可以更方便地进行部署。同时，在生成这两个文件的过程中，还会对模型进行常量折叠、Batch 层融合、水平方向层融合、无效节点消除等模型优化操作。

[13] 该数据所使用的测试配置为：处理器：双路英特尔® 至强® 铂金 8280 处理器，2.70GHz；核心/线程：28/56；HT：ON；Turbo：ON；内存：192GB DDR4 2933；硬盘：英特尔® 固态盘SC2KG48；BIOS：SE5C620.86B.02.01.0008.031920191559；操作系统：CentOS Linux 7.6.1810；Linux内核：4.19.5-1.el7.elrepo.x86_64；gcc版本：4.8.5；OpenVINO™ 工具套件：2019 R1；工作负载：Dense U-Net。

如图 2-1-15 所示，英特尔® DLDT 可以轻松地导入 GE 医疗基于 TensorFlow 等框架训练得到的模型。

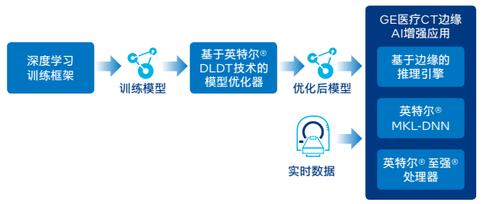


图 2-1-15 部署有英特尔® DLDT 的 GE 医疗 CT 边缘 AI 增强应用

利用英特尔® DLDT 对模型进行转换和优化后，可将优化后的模型导入 GE 医疗 CT 边缘 AI 增强应用中，该应用在英特尔® 至强® 处理器和英特尔® MKL-DNN 的基础上，构建了基于边缘的强大推理引擎。

为了验证这一优化方案的实际效能，双方进行了一系列的性能测试，该数据集具有 8,834 个 CT 扫描图像。GE 医疗希望在对模型实施优化后，能够在使用小于 4 个处理器核心的情况下，使推理引擎每秒可处理的图像数量达到 100 张。

测试结果显示，在只启动单核心的英特尔® 至强® 处理器 E5-2650 v4 上，优化后的模型可使推理吞吐量提高到优化前的 14 倍。同时，英特尔® 至强® 处理器的多核心性能，使得 GE 医疗推理引擎的效率获得大幅提升，如图 2-1-16 所示，在使用了 4 个处理器核心后，推理引擎每秒可处理的图像数量提升到了 596 张，近 6 倍于最初的期望值。[14]

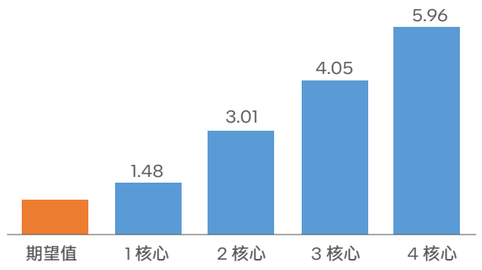


图 2-1-16 多核心带来了推理性能的稳步扩展

汇医慧影利用英特尔技术，构建高效协助诊疗平台

■ 背景

医学上，可以通过超声波、X 光检测、核磁共振成像以及其他医学影像技术来进行辅助诊断。前文也提到，影像的判读需要医生具有丰富的经验以及横跨多学科的知识储备。拥有这些技能的医生，即便在一些大型医院也数量不足，而小型社区医院或边远地区医疗机构则更为稀缺。

同时，虽然医疗影像数量的增长与计算机图像技术的成熟，推动了计算机医疗影像分析解决方案的出现，但由于传统图像诊断支持系统的准确率达不到人工识别的水平，所以医生往往只会用其作为分析诊断前的单一筛查分类和预判断。另外，由于缺乏统一的数据互联互通标准，在面对治疗期内同一患者由多位医生诊治的场景时，也会带来沟通成本上升。

为帮助医疗机构获得更具效能的智能化辅助诊疗平台，作为以人工智能赋能分级诊疗和精准医疗为使命的高新技术企业，汇医慧影与英特尔展开深入合作，通过引入 OpenVINO™ 工具套件以及其他先进软硬件产品，构建基于深度学习方法的辅助诊疗解决方案（Dr. Turing AI），并已在一些疾病的早期筛查和诊断等应用中，获得了令人满意的效果。

■ 方案与成效

作为全新基于深度学习方法的智能图像辅助诊断方法，Dr. Turing AI 新方案可以运用于病灶早期筛查和诊断的全流程，并以统一良好的数据连通性，帮助医务人员提高图像分析、诊断、临床检测支持及疾病管理效率，显现多项优势：

- 影像分析更为准确，并提供多种自动标识能力；
- 图像辅助分析速度更快，提升医生阅片效率；
- 提供基于美国放射学会（ACR）标准的结构化图像报告；
- 可在病灶图像报告和数据系统中自动更新患者信息。

为获得更高的影像分析准确率，方案可以根据需要使用多种深度学习算法模型，如 Inception V4、Inception ResNet V2 等。在最新的一些应用中，如图 2-1-17 所示，方案采用了以 ResNet50 卷积网络模型为基础网络（Backbone）的 RetinaNet 目标检测模型，来实施模型训练及推理，其中 ResNet50 卷积网络模型用于提取特征，子网络用于分类和回归。

[14] 该数据所使用的测试配置为：处理器：英特尔® 至强® 处理器 E5-2650 v4，2.20GHz；核心 / 线程：12/24；HT：ON；Turbo：ON；内存：264GB；硬盘：480GB；操作系统：CentOS Linux 7.4.1708；Linux 内核：3.10.0-693.el7.x86_64；gcc 版本：4.8.5；工作负载：包含了 8,834 个 CT 扫描图像的数据集。

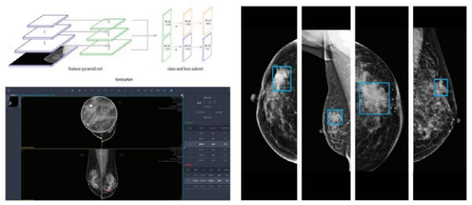


图 2-1-17 基于 RetinaNet 模型构建的方案

为进一步提高分析速度，新方案还引入了 OpenVINO™ 工具套件来提升推理性能。一方面，OpenVINO™ 工具套件中内置的一系列优化工具和预训练模型，可供用户调用，并对已训练完成的模型进行压缩和加速，进而提升模型推理效率；另一方面，方案也能使用 OpenVINO™ 工具套件完成 FP32 模型到 INT8 模型的转换，以可控的模型精度损失换取推理速度的大幅提升（以图像分类为例，业界通用的模型精度损失为小于 1%）。

方案的训练过程采用了精度较高的 Keras FP32 浮点类型模型，而在之后的推理过程中，则使用 OpenVINO™ 工具套件中的模型优化器（Model Optimizer）将原始模型转换为 IR 文件，并输入推理引擎（Inference Engine）中进行推理，再利用其内置的量化工具（Calibration Tool），将 FP32 模型量化为 INT8 类型来提高推理速度。

如图 2-1-18 所示，采用 OpenVINO™工具套件对 FP32 模型进行推理，速度是原始模型的 3.02 倍，而采用 OpenVINO™ 工具套件进行 INT8 转换后，更是将推理速度提升至 8.24 倍，且精确度只损失了不到 0.17%[15]。

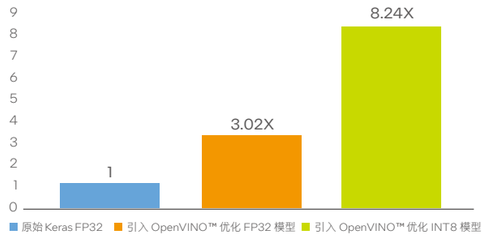


图 2-1-18 OpenVINO™ 工具套件带来的推理效率优化

卫宁健康基于英特尔先进产品，构建高效的智能辅助诊断系统

■ 背景

在肺部疾病的临床诊断工作中，CT 影像不仅是重要的诊断依据，也给拟定治疗方案提供了关键信息。

将 AI 引入智能辅助诊断，可以帮助医疗机构有效应对这一挑战。为此，卫宁健康科技集团股份有限公司（以下简称：卫宁健康）与英特尔和 AMAX 一起，基于深度学习方法，构建了全新的智能辅助诊断系统。系统中的智能辅助诊断模型与放射信息管理系统（Radiology Information System, RIS）和影像归档和通信系统（Picture Archiving and Communication Systems，PACS）相互连通，可将相关影像学定量的表现插入 RIS 报告中，并通过三维智能重建，展现病理组织同周围组织、血管的关系，能更有效地辅助医生观察疑似症状。

为使系统具有更优的部署和运行效能，卫宁健康选择了基于第二代英特尔® 至强® 可扩展处理器，以及内置 OpenVINO™ 工具套件的 AMAX 深度学习一体机做为基础设施。新的处理器不仅拥有强大的通用计算能力，还集成了英特尔® AVX-512、英特尔® DL Boost 等创新技术，能够很好兼顾通用计算能力和并行计算能力，为人工智能训练提供了卓越的性能。而 OpenVINO™ 工具套件包含了大量由英特尔调优和封装的预训练模型，便于用户直接调用。同时，用户还可使用 OpenVINO™ 模型转换器进行数值类型转化来提升效率（详见 1.3.2 节描述）。

如图 2-1-19 所示，在后续的分割、检测、去假阳性这三种任务场景中的测试数据表明，OpenVINO™ 工具套件可将推理速度提升 10-30 倍[16]。

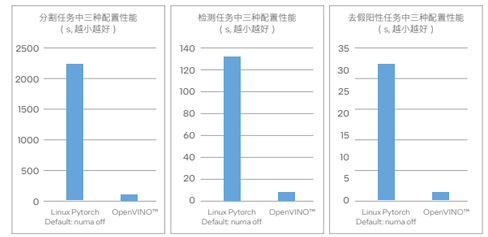


图 2-1-19 智能辅助诊断系统在不同任务场景中的表现

[15] 数据援引自汇医慧影内部测试数据 :https://builders.intel.com/docs/aibuilders/huiying-medical-technology-optimizes-breast-cancer-early-screening-and- diagnosis-with-intel-ai-technologies.pdf，所使用的测试配置为：处理器：双路英特尔® 至强® 铂金 8268 处理器，2.90GHz；核心 / 线程 :24/48；OpenVINO™ 工具套件版本为英特尔发行版 2019R2。

[16] 相关测试配置：双路英特尔® 至强® 金牌 6240 处理器、18 核 /36 线程、启用超线程技术；总内存：384 GB（12 插槽 /32GB/2666MHz）；存储：英特尔® 固态盘 D3-S4510；BIOS：SE5C620.86B.02.01.0010.010620200716（ucode: 0x400002C），CentOS 8，Kernel: 5.6.4-1.el8.elrepo.x86_64；深度学习框架：PyTorch；编译器：gcc 7.3；MKL DNN 版本：v0.20.5；精度：FP32，数据集：357x4x3x96x512x512；定制 3D U-Net；配置 1：Linux PyTorch（1.3.0）Default Numa OFF，1 实例；配置 2：Linux PyTorch（1.3.0）Optimized Numa ON，36 实例；配置 3：OpenVINO，版本：2019.3.376。

致远慧图借力英特尔技术，推出智能远程阅片方案

在传统医疗信息系统中，医院会将采集到的医学影像暂存到图像仓库（ImageHub），然后上传到云端服务器上进行分析处理，再将处理结果返回到医院的应用软件上，帮助医生进行疾病诊断。如图 2-1-20 所示，在这一过程中，结果的反馈速度可能受到网络因素以及推理速度的制约，影响诊疗效率。

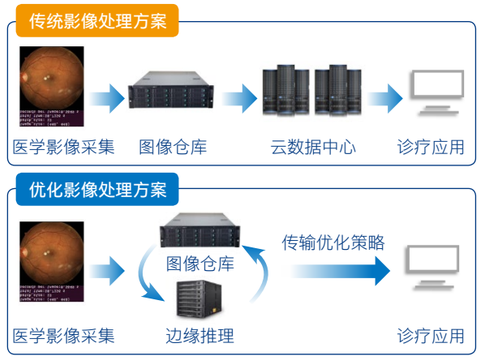


图 2-1-20 智能远程阅片新旧方案对比

为解决这一问题，北京致远慧图科技有限公司（以下简称“致远慧图”）一方面通过架构优化，如图 2-1-20 所示，在贴近医疗一线的边缘侧部署“英特尔® Movidius™ 神经计算棒 +OpenVINO™ 工具套件”，来充分前置 AI 推理能力，让方案在边缘侧就完成模型的压缩、加速和推理过程，降低网络传输带来的延迟。

另一方面，在医学影像分析场景常用的深度学习模型中，采用 INT8 等低精度定点计算方式，可以更高效地利用高速缓存，减少带宽瓶颈，并更大限度地利用处理器计算资源，提升模型的推理速度。因此，致远慧图充分运用英特尔® 架构的处理器特性，借助 OpenVINO™ 工具套件实施模型优化。

如图 2-1-21 所示，OpenVINO™ 工具套件会将训练好的模型（假设使用 PyTorch 框架）通过 PyTorch 提供的工具转换为 ONNX 模型，再使用模型优化器将其转化为 OpenVINO™ 工具套件独有的优化中间表示文件（Intermediate Representation，IR），其包括了 bin 和 xml 两种格式的文件；尔后 Calibrate 工具会使用标注数据集，对模型进一步量化。

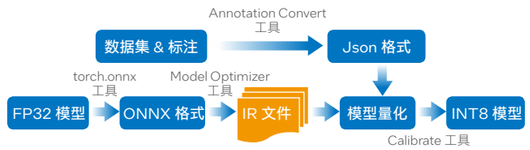


图 2-1-21 借助 OpenVINO™ 工具套件开展模型优化

假设使用 ResNet34 模型，输入图像分辨率为 256*256；任务是 OCT 上的 NORMAL、CNV、DME、DRUSEN 四分类任务。首先使用 torch.onnx 工具，将模型转化为 ONNX 格式的代码示例如下：

```
SIZE = 256
dummy_input = torch.randn (8, 3, SIZE, SIZE)
model = net._dict__['resnet34'] (pretrained = True, num _ classes = 4)
model.load_state_ dict(torch.load('model/resnet34.pth', map_location = 'cpu'))
model.eval( )
torch.onnx.export(model, dummy_input, "model/resnet34.onnx", verbose = True)
```

使用 Model Optimizer 工具，生成 IR 文件命令如下：

```
python mo_onnx.py --input_model model/resnet34.onnx --data_type FP32 --
output_dir model/FP32 --input_shape [1,3,256,256] --scale 255
```

执行结果如图 2-1-22 所示，此时 IR 文件是 FP32 格式，包括了 resnet34.xml 和 resnet34.bin 两个文件。

```
Model Optimizer arguments:
Common parameters:
        - Path to the Input Model:      /home/jie.wang/code/intel_ncs/model/resnet3
4.onnx
        - Path for generated IR:        /home/jie.wang/code/intel_ncs/model/FP32
        - IR output name:       resnet34
        - Log level:    ERROR
        - Batch:        Not specified, inherited from the model
        - Input layers:         Not specified, inherited from the model
        - Output layers:        Not specified, inherited from the model
        - Input shapes:         [1,3,256,256]
        - Mean values:  Not specified
        - Scale values:         Not specified
        - Scale factor:         255.0
        - Precision of IR:      FP32
        - Enable fusing:        True
        - Enable grouped convolutions fusing:   True
        - Move mean values to preprocess section:       False
        - Reverse input channels:       False
ONNX specific parameters:
Model Optimizer version:        2019.1.1-83-g28dfbfd

[ SUCCESS ] Generated IR model.
[ SUCCESS ] XML file: /home/jie.wang/code/intel_ncs/model/FP32/resnet34.xml
[ SUCCESS ] BIN file: /home/jie.wang/code/intel_ncs/model/FP32/resnet34.bin
[ SUCCESS ] Total execution time: 6.03 seconds.
```

图 2-1-22 使用 Model Optimizer 工具生成的 IR 文件

接下来需要将模型进一步量化，这需要准备与任务相关的数据集以及标注，并使用 OpenVINO™ 工具套件提供的 Annotation Convert 工具将数据集转换为标准格式。因为本文假设的是多分类任务的模型，因此使用 imagenet 格式组织数据，然后使用工具进行转化。如图 2-1-23 所示，数据集的组织格式，从左到右分别是图像文件夹、图像标注及标注对应名称。

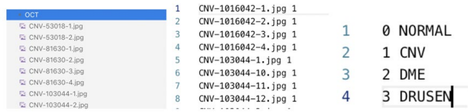


图 2-1-23 数据集的组织格式

标注转化命令如下：

```
python accuracy_checker_tool/convert_annotation.py imagenet --annotation_file labels.txt --
labels_file synset_words.txt -o datasets/annotations/OCT/ -a oct.pickle -m oct.json
```

转化完成后，可以得到一个 json 文件：

```
oct.json:
{
  "label_map": {
    "0": "NORMAL",
    "1": "CNV",
    "2": "DME",
    "3": "DRUSEN"
  }
}
```

此时，需要借助 OpenVINO™ 工具套件提供的 Calibrate 工具对模型进一步量化，将模型从 FP32 量化为 INT8，来进一步提升模型的推理速度。本文中采用的 resnet34.yml 中包括模型的定义和权重、模型的任务类型，以及使用的框架、使用的数据集等。文件如下所示：

```
models:
  - name: resnet34
    launchers:
      - framework: dlsdk
        device: CPU
        model: resnet34.xml
        weights: resnet34.bin
        adapter: classification
        cpu_extensions: AUTO
    datasets:
      - name: OCT
```

使用 definition.yml 定义 launchers 的框架和设备，以及各种数据集的地址、标注和评价指标，这里使用 accuracy 的 top1 评价指标。文件如下所示：

```
launchers:
  - framework: dlsdk
    device: CPU
datasets:
  - name: OCT
    data_source: datasets/OCT
    annotation: oct.pickle
    dataset_meta: oct.json
    metrics:
      - name: accuracy @ top1
        type: accuracy
```

Calibrate 工具量化命令如下：

```
python calibrate.py --config resnet34.yml -d  definitions.yml -
M ~/intel/openvino/deployment_tools/model_optimizer --source  ~/code/intel_ncs/ -
annotations  ~/code/intel_ncs/ --models ~/code/intel_ncs/model/FP32
```

后续进行的验证测试结果表明，借助 OpenVINO™ 工具套件，AI 应用能更充分挖掘基于英特尔® 架构的处理器的计算资源。且经进一步转化为 INT8 模型后，在基本不影响准确率的情况下，推理速度能获得显著提升，有效地缩短了影像处理的响应时间，能够帮助医疗机构提高诊疗效率。

小结

医疗图像分割、目标检测是 AI 应用于医疗方向的重要分支。良好的图像分割模型，能有效帮助医疗机构提高医学影像判读效率，进而增强临床诊疗能力以及减少病患等待时间，弥补因医疗机构影像科资源缺乏带来的多种问题。

与基于 AI 在其他图像处理领域的应用不同，医疗领域的图像分割对时效性要求更高，留给病患的黄金诊疗窗口往往只有数十分钟。因此，如果图像分割 AI 应用的推理效率不够高，就有可能延误宝贵的抢救时间。来自多个行业、多个场景的案例显示，英特尔® 至强® 可扩展处理器、第二代英特尔® 至强® 可扩展处理器，以及英特尔® 深度学习加速指令集、OpenVINO™ 工具套件等产品和技术，可以有效提升深度学习模型的推理效率。基于不断创新的产品与技术，英特尔也将一如既往地推动医疗行业中 AI 应用的创新和落地，使科技更好地服务于人们的健康生活。

AI + Cloud，协力共建高效医学影像分析能力

医疗领域中的医学影像分析

医学影像分析面临挑战

众所周知，高水平诊疗的前提，是对病情的准确掌握和精准分析。古时，医技高明的大夫以望、闻、问、切来获取和推断病情。今天，通过各类医疗设备和信息系统，尤其是医学影像设备的辅助，医生更能驾驭诊疗过程，为病人提供优质医疗服务。目前，在大中型医疗机构中，X 光机、CT 机、核磁共振等设备已逐渐普及，即便在基层医疗机构，患者也能进行各类医学影像检查。

医学影像设备和系统虽然可以迅速到位，但“软实力”却无法一蹴而就。如医学影像分析需要影像科医生拥有较高的专业素养，不仅具备临床医学、医学影像学等方面的专业知识，还必须熟练掌握放射学、CT、核磁共振、超声学等相关技能，同时，还需具备运用各种影像分析技术进行疾病诊断的能力。

因此，虽然医学影像设备在医疗机构已相当普及，但在一些边远地区或基层医疗机构，却常常面临空有设备却无人有能力“看片”的尴尬境地。以一些省份为例，很多医学影像设备已部署到县、社区一级的医疗机构，但病人接受检查后，当地医院却依然无法做出精准的判断和分析，需要将影像文件通过拍照、扫描等方式传给上一级医疗机构。有时会因为影像文件的质量得不到保障乃至失真，造成病情的延误或误判。

不仅如此，由于各医疗机构的信息化系统彼此独立，且数据标准未完全统一。例如各个 PACS 上存储的医学影像数据几乎没有连通，形成了一个个信息“孤岛”，这些都会造成偏远地区患者在基层医疗机构得不到有效的病情分析，长途奔波到大医院后，却还需要接受重复检查的怪现象，存在引发医患矛盾的风险。

“云技术+大数据”在医学影像分析中的应用

云计算技术的快速发展，让信息孤岛问题逐渐得以解决，如图 2-2-1 所示，越来越多的医疗机构开始将相关医技设备及医疗服务过程都通过云的方式链接起来，并在其上构建全医技协同平台、影像协同平台等能力和应用，以平台即服务（Platform as a Service，PaaS）或软件即服务（Software as a service，SaaS）的方式满足各层级医疗机构的不同需求。

以全医技协同服务平台为例，通过接入云服务，各级医疗机构能够获得跨终端、跨平台的全医技功能应用。而影像协同平台则能够让来自大、中型医疗机构的医学影像专家随时随地处理从不同地区传来的影像数据，并对疑难杂症进行协同会诊，来实现医疗资源的高效共享。

以医学影像数据为例，基于云计算和大数据技术的互联互通，不仅让各医疗机构可以规避过度检查、重复治疗等问题，还有力地打破了数据孤岛现象，建立起无边界医疗全连接，提高了医疗服务质量。同时，通过影像数据的积累和分析，也让基于 AI 的医学影像分析应用日趋走向成熟。现在，基于云技术+AI 的医学影像分析已逐渐在各个医疗机构获得部署，并获得良好反馈。

基于 AI 的医学影像分析

通过云服务和大数据系统汇集的海量数据，让目标侦测神经网络等 AI 模型获得大量的训练样本，令基于 AI 的智能化辅助诊断系统能够更有效地帮助医疗机构提升诊疗能力。

现在，在医学影像 AI 分析应用中，部分医疗机构正利用低剂量 CT 对病灶进行智能化辅助诊断。实践数据显示，其定量的监测敏感度（探测率）已达到 95%，筛查时间也由人工所需

图 2-2-1 云服务将医技设备聚合起来

的 10 多分钟缩短到秒级[17]。通过 AI 模型识别出病灶后，再交由医生执行进一步诊断，效率和精准度都获得了大幅提升。

目前，在医学影像 AI 分析应用中，目标侦测神经网络正被广泛地运用，其通过深度学习的方法，能够对 X 光片、CT 成像等医学影像进行高效、准确的病灶检测。

目标侦测神经网络

典型的目标侦测神经网络有 R-CNN、Fast R-CNN、SPP-NET、R-FCN[18] 等。R-FCN 是近年来在医学影像分析领域常见的目标侦测神经网络模型。

一个典型的 R-FCN 结构，如图 2-2-2 所示，首先，对需要处理的影像图片进行预处理操作后，送入一个预先训练好的卷积神经网络（CNN）中，例如 ResNet-101 网络。在该网络最后一个卷积层获得的特征地图（feature map）上，会引出 3 个分支。第 1 个分支是将特征地图导入区域生成网络（Region Proposal Network，RPN），并获得相应的兴趣区域（Region Of Interest，ROI）；第 2 个分支是在该特征地图上获得一个用于分类的多维位置敏感得分映射（position-sensitive score map）；第 3 个分支就是在该特征地图上获得一个用于回归的多维位置敏感得分映射。最后，在两个位置敏感得分映射上，分别执行位置敏感的 ROI 池化操作（Position-Sensitive ROI Pooling），由此获得对应的类别和位置信息。

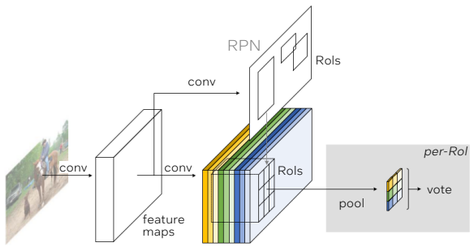


图 2-2-2 典型的 R-FCN 结构

与其他目标侦测神经网络模型，例如 Faster R-CNN 相比，R-FCN 具有检测速度更快，检测精度也更高等特点[19]。

软硬件配置建议

对于基于 AI 的医疗影像分析方案构建，可以参考以下基于英特尔® 架构平台的软硬件配置来完成。

名称	规格
处理器	英特尔® 至强® 金牌 6240 处理器或更高
超线程	ON
睿频加速	ON
内存	16GB DDR4 2666MHz* 12 及以上
存储	英特尔® 固态盘 D5 P4320 系列及以上
操作系统	CentOS Linux 7.6 或最新版本
Linux 核心	3.10.0 或最新版本
编译器	GCC 4.8.5 或最新版本
Caffe 版本	面向英特尔® 架构优化的 Caffe 1.1.6 或最新版本

优化 AI 模型效率

基于英特尔® 架构平台的优化

包括英特尔® 至强® 可扩展处理器、第二代英特尔® 至强® 可扩展处理器等在内的英特尔® 架构平台，不仅可为基于 AI + Cloud 的智能医疗影像分析系统带来强大的通用计算能力，更可为其提供亟需的并行计算能力。在深度学习模型的推理过程中，往往对并行计算能力有着较高要求，而英特尔® 至强® 可扩展处理器通过引入英特尔® AVX -512，提供了更高效的单指令多数据流（Single Instruction Multiple Data，SIMD）执行效率，让系统获得了更强大的并行计算加速能力。

同时，英特尔® 数学核心函数库（Intel® Math Kernel Library，英特尔® MKL）、英特尔® MKL-DNN 的加入，可以进一步提升 AI 模型的工作效率，其主要通过以下三个方面来提升人工智能模型性能：

- 使用 Cache Blocking 技术优化数据缓存，提高数据命中率；
- 对神经网络中的常用算子进行并行化与向量化优化；
- 使用 Winograd 算法级优化。

而全新的第二代英特尔® 至强® 可扩展处理器中加入的英特尔® 深度学习加速技术，让深度学习推理可以使用 INT8 来获得更佳的性能表现。

[17] 数据援引自盈谷内部测试数据：https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/yinggu-case-study-medical.html
[18] R-FCN 相关技术描述，援引自 Jifeng Dai, Yi Li, Kaiming He, Jian Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks, https://arxiv.org/pdf/1605.06409v2.pdf
[19] R-FCN 性能数据，请参阅 Jifeng Dai, Yi Li, Kaiming He, Jian Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks, https://arxiv.org/pdf/1605.06409v2.pdf

在英特尔® 至强® 可扩展处理器平台上，以单幅胸部 Dicom 数据执行 R-FCN 模型为例，来自某应用的数据表明，英特尔® 至强® 金牌 6148 处理器经过优化，可以把性能提升近 5 倍[20]。

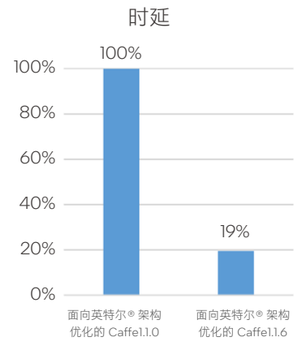


图 2-2-3 单幅胸部 Dicom 数据执行 R-FCN 模型处理比较

面向英特尔® 架构优化的Caffe

与伯克利视觉和学习中心（Berkeley Vision and Learning Center，BLVC）版本的 Caffe[21] 相比，面向英特尔® 架构优化的 Caffe[22] 专门面向英特尔® 架构进行了大量优化，并加入了对英特尔® MKL、英特尔® MKL-DNN 以及英特尔® AVX-512 的支持，在各个深度学习模型上都有着更好的性能表现，推理效率也更高。

为了使英特尔® 架构处理器的计算资源得以充分利用，一般在执行推理之前还可以进行一些环境变量的设置，例如：

```
export OMP_NUM_THREADS=36
```

这里 OMP_NUM_THREADS 是指定要使用的线程数。

通过对 BLVC Caffe 实施的性能分析，面向英特尔® 架构优化的 Caffe 进行了以下几个方面的优化。

■ 代码矢量化优化

优化内容包括：

- 将基本线性代数子程序（BLAS）库从自动调优线性代数系统（ATLAS）切换至英特尔® MKL-DNN，从而使通用矩阵乘法（GEMM）等优化后，更适用于矢量化、多线程化的工作负载，并提高缓存量；
- 使用 Xbyak just-in-time（JIT）汇编程序执行编译过程。作为一种 x86/x64 JIT 汇编程序，Xbyak 对英特尔® 架构下的指令集，例如 MMX™ 技术、英特尔® 流式单指令多数据扩展（Intel® Streaming SIMD Extensions, 英特尔® SSE）、英特尔® AVX 系列技术等有着更好的支持；同时，还可帮助面向英特尔® 架构优化的 Caffe 在代码实施过程中提高矢量化率；
- 对 GNU Compiler Collection（GCC）和 Open Multi-Processing（OpenMP）进行代码矢量化。矢量化率的提高，有利于 SIMD 指令同时处理更多数据，提高数据并行利用率。同时，对代码进行矢量化处理，也能有效提升深度学习模型中池化层的性能。

■ 常规代码优化

优化内容包括：

- 降低编程复杂性；
- 减少计算数量；
- 展开循环。

例如在代码优化过程中采用一些标量优化技巧，代码如下：

```
for (int h_col = 0; h_col < height_col; ++h_col) {
  for (int w_col = 0; w_col < width_col; ++w_col) {
    int h_im = h_col * stride_h - pad_h + h_offset;
    int w_im = w_col * stride_w - pad_w + w_offset;}}
```

其代码片段的第三行，关于 h_im 计算，可以将其移出最内层，如下所示：

```
for (int h_col = 0; h_col < height_col; ++h_col) {
  int h_im = h_col * stride_h - pad_h + h_offset;
  for (int w_col = 0; w_col < width_col; ++w_col) {
    int w_im = w_col * stride_w - pad_w + w_offset;}}
```

■ 基于英特尔® 架构处理器的其他优化措施

优化内容包括：

- 改进 im2col_cpu/col2im_cpu 执行效率，im2col_cpu 函数是深度学习计算中的常用函数，其能使用优化后的 BLAS 库，以 GEMM 方式执行直接卷积。可对 im2col_cpu 实施以下优化：在 BLVC Caffe 代码中

```
for (int c_col = 0; c_col < channels_col; ++c_col)
  for (int h_col = 0; h_col < height_col; ++h_col)
    for (int w_col = 0; w_col < width_col; ++w_col)
      data_col[(c_col*height_col+h_col)*width_col+w_col] = // ...
```

[20] 性能测试结果基于【2019 年 4 月 10 日】进行的测试，测试配置为：2 路英特尔® 至强® 金牌 6148 处理器，20 核心 /40 线程，启用 HT/Turbo，搭载 192GB 内存（12 slots / 16GB / 2666MHz），CentOS 7.6, BIOS:SE5C620.86B.02.01.0008.031920191559（uncode:0x200005e），Kurnel 版本 : 3.10.0-957.21.3.el7.x86_64，编译器 GCC 4.8.5。测试组使用英特尔® MKL-DNN 0.12 版本，对比组使用英特尔® MKL-DNN 0.18 版本，框架：面向英特尔® 架构优化的 Caffe1.1.0，对比组使用面向英特尔® 架构优化的 Caffe 1.1.6. Minibatch=1 配置下完成。
[21] 该版本源代码请详见https://github.com/BVLC/caffe
[22] 该版本源代码请详见https://github.com/intel/caffe

其中的四次算术运算（两次加法和两次乘法），可替换为单次索引递增运算来提升运算效率；

- 降低归一化批处理的复杂性；
- 特定的处理器 / 系统的优化方法；
- 每个计算线程锁定一个核心，避免线程移动，可设置如下环境变量来实现。

```
export KMP_AFFINITY=granularity=fine,compact,1,0
```

通过紧密设置相邻线程，可提高 GEMM 操作性能，因为所有线程都可共享相同的末级高速缓存（LLC），从而可将之前预取的缓存行重复用于数据，提高效率。

■ 借助 OpenMP 实现代码并行化

采用 OpenMP 多线程并行处理方法，可以有效提升神经网络的推理效率，例如在池化层中，单一池化层适用于处理单张特征图，但如果池化层与 OpenMP 多线程并行执行，由于图像相互独立，因此多个线程可并行同时处理多个图像，提升效率。代码如下：

```
#ifdef _OPENMP
 #pragma omp parallel for collapse(2)
#endif
 for (int image = 0; image < num_batches; ++image)
  for (int channel = 0; channel < num_channels; ++channel)
   generator_func(bottom_data, top_data, top_count, image, image+1,
                  mask, channel, channel+1, this, use_top_mask);
}
```

可以看出，借助 collapse（2）clause，OpenMP #pragma omp parallel 可以扩展到两个 for-loop 嵌套语句，再将批量迭代图像和图像通道两个循环合并成一个循环，并对该循环进行并行化处理。

通过一系列的优化方法和技巧，面向英特尔® 架构优化的 Caffe 在性能上相较 BLVC Caffe 有了长足的提升。一项测试表明，面向英特尔® 架构优化的 Caffe，工作负载执行时间可缩短至原来的 10%，而整体执行性能则提升到原来的 10 倍以上 [23]。

* 更多面向英特尔 ® 架构优化的 Caffe 的技术细节，请参阅本手册技术篇相关介绍。

[23] 相关测试数据，以及更多面向英特尔® 架构优化的Caffe的优化方法，请参阅《Caffe* Optimized for Intel® Architecture: Applying Modern Code Techniques》：https://software.intel.com/en-us/articles/caffe-optimized-for-intel-architecture-applying-modern-code-techniques。

西安盈谷利用 AI 技术和云服务，提升医学诊疗辅助能力

背景

医疗资源配置的不均衡，使各个医疗机构在医疗影像的后处理、后分析能力上也参差不齐。同时，数据没有互联互通，也使医疗资源的利用效率难以通过资源共享得到有效提升。专注医学影像核心技术近 20 年的西安盈谷网络科技有限公司（以下简称“西安盈谷”），正致力于将其专业医学影像核心技术和产品，与最新的云计算、大数据和 AI 技术结合起来，形成高效、智能的医疗智能化辅助诊断能力，助力广大医疗机构提升诊疗效率及质量。

在西安盈谷看来，要解决医学影像分析处理能力发展不均衡的问题，就必须通过云计算等方式将医学影像数据有效聚合起来，并在其上形成基于 AI 的数据分析能力，进而以资源共享和 AI 两大能力，来逐渐消除各级医疗机构在医学影像分析能力上的差异。

为此，西安盈谷通过医真云的部署，利用创新的医技设备物联网技术 AMOL，将源自不同设备的海量医学影像数据链接起来。同时，西安盈谷还将深度学习引入医学影像处理中，基于目标侦测神经网络模型构建了全新的 Cloud IDT 服务，在提高检出率、降低决策时间、提高工作效率等多个方面都收效显著。

为帮助西安盈谷更好地推动这一系统的部署落地，英特尔为其提供了英特尔® 至强® 可扩展处理器等最新一代平台产品与技术，助其完成了 Cloud IDT 服务向英特尔® 架构平台的迁移，以及对于 Caffe、TensorFlow 等深度学习框架的部署和优化。通过双方的协作和努力，全新的医疗智能化辅助诊断系统已经在筛查时间、准确率等多个指标维度上获得了用户的一致好评。

方案与成效

在新方案中，一方面，西安盈谷基于目标侦测神经网络模型构建了一系列医学影像分析处理应用，并采用英特尔® 架构处理器执行高效率的模型推理；另一方面，西安盈谷也将其 Cloud IDT 智能应用与医学影像处理及分析云计算 @ iMAGES 核心引擎等结合起来，提供了强劲的影像大数据在线智能处理能力。

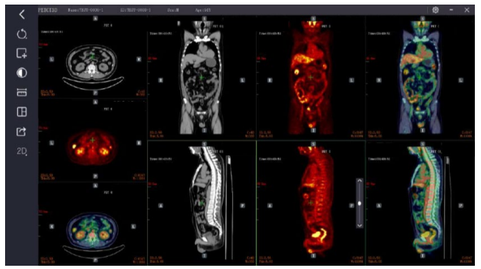


图 2-2-4 云端 PET-CT 融合

如图 2-2-4 所示，结合基于英特尔® 架构的处理器提供的强劲算力，以及 @iMAGES 核心引擎提供的基于云端的强大正电子发射计算机断层显像（Positron Emission Tomography CT，PET-CT）融合能力，不仅能够提供基于形态学和功能的“热力图”，还可以对影像做出半定量化的标准化摄取值（Standard Uptake Value, SUV）分析，而这些影像又可通过 Cloud IDT 智能系统中的 R-FCN 目标侦测神经网络，进一步执行疾病的鉴别和定量分析。

在出色的硬件性能基础上，英特尔还通过对 Caffe、TensorFlow 等人工智能框架的优化，进一步提升了西安盈谷 Cloud IDT 智能系统的执行效率。通过对 R-FCN 模型的优化，模型裁剪融合带来了近 30% 的性能提升，而进一步优化 OpenMP 多线程实现方案后，性能再度提升 40-50%[24]。

此外，英特尔® 至强 ® 可扩展处理器在通用计算能力和并行计算能力两方面的算力支撑，也可助力智能系统将原先分散在不同平台的任务处理，例如数据统计与模型推理，合并到一起，进而让用户不仅能在其私有云中部署更多的虚拟机，还能降低总拥有成本（Total Cost of Ownership, TCO）。

[24] 数据援引自盈谷内部测试数据：https://www.intel.cn/content/www/cn/zh/analytics/artificial-intelligence/yinggu-case-study-medical.html，所使用的测试配置为：处理器：双路英特尔® 至强® 金牌 6148 处理器，2.40GHz；核心/线程：20/40；HT：ON；Turbo：ON；内存：192GB DDR4 2666；硬盘：英特尔® 固态盘 SC2KB48；网络适配器：英特尔® 以太网聚合网络适配器 XC710；BIOS：SE5C620.86B.02.01.0008.031920191559；操作系统：CentOS Linux 7.6；Linux内核：3.10.0-957.21.3.el7.x86_64；gcc版本：4.8.5；Caffe版本：面向英特尔® 架构优化的Caffe 1.1.6；工作负载：R-FCN。

小结

以数据驱动医疗信息化的美好明天，是英特尔与西安盈谷等合作伙伴的共同心愿。基于云计算、物联网、大数据以及 AI 等技术领域，针对医疗信息化、智能化的应用目前已经得到了广泛的开展和探索，并在医学影像数据实时计算展现、医学视觉类数据人工智能研究等多个方面都获得了突破，在各个医疗机构的实际部署和实施中都获得了良好的反馈。

为不断挖掘目前主流 AI 框架在基于英特尔® 架构的平台上的潜力，英特尔对这些框架开展了多方位的优化工作。面向英特尔® 架构优化的 Caffe 框架通过代码矢量化、借助 OpenMP 并行化等优化手段，使模型整体性能相较 BLVC Caffe 获得巨大提升，在与西安盈谷 Cloud IDT 智能应用、医学影像处理及分析云计算 @ iMAGES 核心引擎等应用结合后，已在一大批关键场景中建立起 “AI+Cloud” 的智能辅助诊断系统能力。随着第二代英特尔® 至强® 可扩展处理器、英特尔® 傲腾™ 持久内存等英特尔技术与产品的涌现，相信基于英特尔® 架构平台构建的医疗影像分析解决方案会输出更强大的性能表现以及更高超的 AI 能力。未来，英特尔还计划与更多合作伙伴继续深入开展合作，将更多、更先进的产品与技术与医疗信息化进程结合起来，推动精准医疗、智慧医疗的前行，让信息化、数字化和智能化更有效地提升医疗服务水平，为患者带去更舒心和贴心的医疗健康服务。

AI 技术加速病理图像分析

医疗领域中的病理切片分析

传统病理切片分析方法面临挑战

病理切片是将部分病变组织或脏器，经过一系列处理后形成微米级薄片，粘附在玻片上并进行染色处理，然后再交至病理科，病理科医生通过显微镜对病理切片进行镜检，观察病理变化，并作出病理诊断和预后评估。病理切片检查是一项非常复杂和具有挑战性的工作，而想要成为病理学方面的专家，更是需要具备多年的读片经验与数万张切片的阅片积累以及具有丰富专业知识。然而，据统计，目前全国病理科医生还不足万人[25]。

此外，人工检查不免带有较大主观性，由不同病理科医生对同一患者的病理切片作出的诊断，也经常会存在差异，这可能导致误诊、漏诊等现象产生。同时，在实际的病理切片检查中，患者的病理切片以 40 倍的放大倍数进行数字化后，单个病理切片的像素点可能超过百万像素。病理科医生需要连续观察多张百万像素级的图片，并且需要注意到图片里微观区域的异常，不仅费时费力，还容易出现错漏。且较长的阅片时间也会导致病患等待时间长，有可能会造成病情的延误。

基于 AI 的病理切片分析方法

随着基于 AI 的图像处理与分析技术获得巨大进步，各个医疗机构均不遗余力地开展了基于深度学习或机器学习的病理切片分析方法，并取得了良好的成效。例如通过 ResNet50 网络进行的深度学习模型训练，可用于执行高危病的病理组织辨识工作。尽管其得到的病灶预测热学图依然存在噪声等问题，但已经可以像病理科医生一样，以不同的放大倍数来检查病理切片图像。实验表明，医疗机构有可能通过训练一个深度网络模型，使其不仅能够具备专业的检测技术，还能有超快的检测速度和无限的工作时间。

来自纽约大学的一项最新研究成果表明，利用大量数字化病理切片图像训练的 Inception v3 深度学习模型，识别病灶组织和正常组织的准确率已达到 99％[26]。

现在，基于 CNN 的分类算法以及目标侦测算法都已经获得了长足的发展。作为深度学习的代表方法之一，CNN 的典型代表，例如 LeNet、ZFNet、VGGNet 和 ResNet 等，已经被广泛地运用于图像分类、目标定位和图像分析等领域。

■ 分类卷积神经网络

在医疗图像的检测结果中，往往会出现明显的分类情况，例如阴性为正常，阳性为非正常。可以看出，此时检测所期望的结果，是一系列的离散数字，例如 0 或 1，这就构成了一个典型的分类问题。据此可以认为，利用类似二分类的分类算法，CNN 能够有效帮助医疗机构先初步、定性地筛选出有问题的区域或组织，然后再进行定量的分析和判读。

典型的二分类算法，如逻辑回归，是一种广义的线性回归分析模型。以根据病理切片图片来预测患者是否患病为例，假设随着患者年龄的增加，当发现某种细胞超过 x 个即可判定患病，此时，其在数学上就表现为一个阈值为 x 的线性函数，即 y=年龄（n）*a+ 初始值（b），当 y>=x 时，判为患病。

而在实际场景中，这一函数会复杂得多，例如除了年龄以外，异常细胞的大小、状态等也可能成为判断依据，此时，线性函数就会变成一个多元线性函数，例如

y=n*a+m*c+o*d……+b

如前所述，分类问题需要输出一系列离散的结果，因此需要在线性函数上加上一个激活函数，使其输出结果呈离散化。而对于神经网络而言，激活函数的作用是能够给神经网络加入一些非线性因素，使神经网络可以更好地解决较为复杂的问题。常见的激活函数有 Sigmoid 函数、tanh 函数、ReLU 函数等。另外，逻辑回归会采用梯度下降迭代求解的方法，来获取最小化的损失函数。

通常，基于二分类算法的 CNN 图像分类具有以下几个主要模块，如图 2-3-1 所示，包括图像读取与预处理、图像训练、迭代优化和图像预测。其中基于 CNN 的模型训练，由卷积层、池化层以及全连接层等构成，可采用交叉熵损失函数，以及 MBGD 梯度下降算法或 BGD 梯度下降算法。

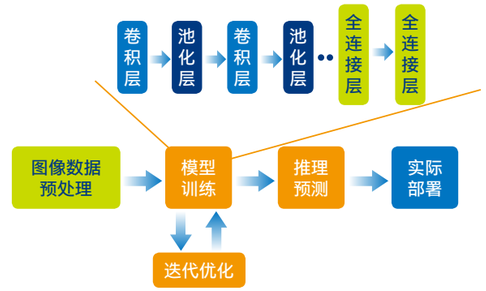


图 2-3-1 基于二分类算法的 CNN 图像分类组成模块

[25] 该数据援引自媒体报道：https://www.cn-healthcare.com/article/20141118/content-463705.html
[26] 数据源引自Coudray N, Moreira A L,Sakellaropoulos T,et al.Classification and Mutation Prediction from Non-Small Cell Lung Cancer Histopathology Images using Deep Learning[J].bioRxiv, 2017.

在实际应用中，残差网络（Residual Net，ResNet）也是常见的分类卷积神经网络之一，其在 2D 图像分类、检测及定位上有着非常优异的特性。与其他 CNN 相比，ResNet 在网络中增加了直连通道，允许输入信息直接传到后面的层中，如图 2-3-2 所示：

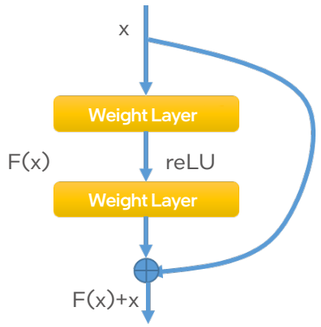


图 2-3-2 ResNet 的残差结构

这一结构（残差结构）在一定程度上解决了经典 CNN 网络结构在信息传递时可能存在的信息丢失、损耗，乃至梯度消失等问题，这些问题是深度模型的层数无法变得太多的原因之一。采用 ResNet 后，训练模型的层数可以大幅增加，也由此提高了分类准确率。

■ 目标侦测神经网络

目标侦测神经网络是指在给定的图片中精确找到物体所在位置，并标注出物体的类别。常见的目标侦测神经网络有 R-CNN、Fast R-CNN、SPP-NET、R-FCN 等。

R-CNN 是经典的深度学习目标检测算法，其基本工作流程如下：首先，R-CNN 会基于 selective search 方法在原始图上生成数千个大小一致的候选区域，并输入 CNN 网络。由该网络模型得到的特征向量将通过多类别的支持向量机（Support Vector Machines，SVM）分类器，每个目标都会训练一个 SVM 分类器，并从特征向量中推断其属于该目标的概率。同时，R-CNN 还设置了一个边界框的回归模型来提升定位准确性，通过边框回归模型对边界框的准确位置进行了优化。

为了解决 R-CNN 在实际应用中训练、推理和测试速度较慢，训练所需空间大等问题，Fast R-CNN 采用了以下方法来应对，并获得了比 R-CNN 更好的应用效果。方法为：

- 将整个图像先进行归一化后再送入 CNN 网络；
- 在卷积层不进行候选区域的特征提取，而是在最后一个池化层加入候选区域坐标信息进行特征提取的计算；
- 在 CNN 网络中统一做目标与候选框回归。

而后续的 Faster R-CNN 又将特征抽取（feature extraction）、proposal 提取，bounding box regression（rect refine）、classification 都整合在了一个网络中，使得综合性能有较大提高，在检测速度方面尤为明显。

软硬件配置建议

对于基于 AI 的病理切片分析方案构建，可以参考以下基于英特尔® 架构平台的软硬件配置来完成。

名称	规格
处理器	英特尔® 至强® 金牌 6240 处理器或更高
超线程	ON
睿频加速	ON
内存	16GB DDR4 2666MHz* 12 及以上
存储	英特尔® 固态盘 D5 P4320 系列及以上
操作系统	CentOS Linux 7.6 或最新版本
Linux 核心	3.10.0 或最新版本
编译器	GCC 4.8.5或最新版本
Caffe 版本	面向英特尔® 架构优化的 Caffe 1.1.6 或最新版本

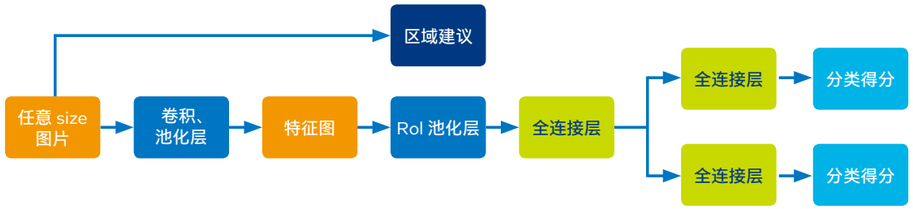


图 2-3-3 Fast R-CNN 网络结构

基于深度学习的病理切片分析方法的优化

基于英特尔® 架构处理器的优化方法

在英特尔® 平台上进行基于深度学习的病理切片分析方法的构建和优化，可以为用户带来以下几个方面的收益：

- 病理切片图像每个文件容量都动辄有数十、上百 MB。传统上，由于存储空间的限制，训练中设定的 Batch Size 都偏小，由此会带来训练时间的增加。而采用基于英特尔® 架构平台，服务器具备了大内存（普遍具备数 TB 乃至数十 TB），可以让 Batch Size 轻松设置至 100 以上，能够加快训练速度；
- 基于 3D XPoint™ 存储介质构建的英特尔® 傲腾™ 持久内存的引入，让至强® 可扩展平台的优势得到进一步加强。与昂贵的动态随机存取存储器（Dynamic Random-Access Memory，DRAM）内存相比，英特尔® 傲腾™ 持久内存大容量和非易失性的特性，及其在实现容量扩展时更低的成本优势，可以有效提升执行模型训练和推理的服务器的内存密度以及计算效率，并大幅降低 TCO；
- 英特尔® 至强® 可扩展处理器创新的微架构，包括更多数量的核心、更高并发度的线程和更充沛的高速缓存，配合它集成的大量硬件增强技术，特别是英特尔® AVX-512 等，都能为 AI 应用提供更强的算力。

面向英特尔® 架构优化的 Caffe

Caffe 是一种常用的深度学习框架，其在视频、图像处理等领域的 AI 训练和推理上有着广泛的运用。为了进一步提升和优化基于 Caffe 的深度学习模型的工作效率，基于英特尔® 架构特性，英特尔对 Caffe 进行了大量优化。

这些优化工作包括：

■ 针对典型 ResNet 网络开展的优化

面向英特尔® 架构优化的 Caffe 利用 ResNet 系列模型特性，来减少计算和内存访问带来的开销。图 2-3-4 是一种典型的 ResNet 的残差结构，从图左半部可以看出，其底部的 2 个 1*1 卷 Stride-2 卷积层只消耗了一半激活操作。优化方案更改了绑定层设置，如图右半部所示，其将一个 1*1 的池化层加入直连通道，减少了一半的计算量。

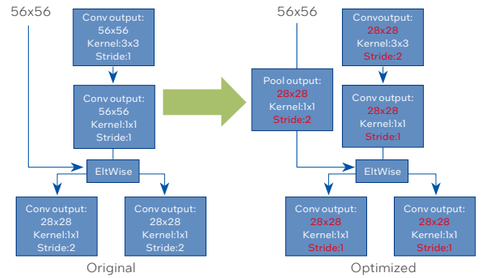


图 2-3-4 面向英特尔® 架构优化的 Caffe 对 ResNet 网络的优化方案

■ 层融合技术

面向英特尔® 架构优化的 Caffe 除了针对指令集的向量化、线程级并行进行优化外，还在 Caffe 框架中引入了更为有效的层融合（Layer Fusion）优化手段，如 BN+Scale、Conv+Sum、Conv+Relu、BN inplace 以及 sparse fusion，这些手段使得神经网络，如 ResNet50 的性能获得了极大提升。如图 2-3-5 所示，这是一种残差结构的 Conv 层与 Eltwise 层的融合，图左半部中的卷积层（Conv）res2a_branch2c 和 Eltwise 层 res2a_relu 被融合到一个新的卷积层 res2a_branch2c 中（图右半部所示），有效地提升了 ResNet 类网络模型的性能表现。

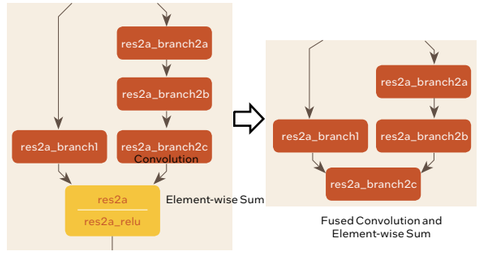


图 2-3-5 Conv 层与 Eltwise 层融合

同时，面向英特尔® 架构优化的 Caffe 还对 INT8 有着良好支持，并提供了 calibration 工具，可帮助用户将神经网络无缝转换到 INT8，以大幅提升性能。

一项测试表明，与使用 BVLC Caffe 相比，面向英特尔® 架构优化的 Caffe 在英特尔® 至强® 可扩展处理器上，通过加入层融合技术，使用 ResNet50 卷积神经网络在同等测评环境中执行 AI 推理，如图 2-3-6 所示，单位时间推理性能可提升达前者的 51 倍之多，推理时长则缩短至前者的 4.7%[27]。

[27] 该数据援引自《Highly Efficient 8-bit Low Precision Inference of Convolutional Neural Networks with Intel Caffe》一文：https://arxiv.org/pdf/1805.08691.pdf，测试配置如下：卷积模型：ResNet50，硬件：AWS single-socket c5.18xlarge。

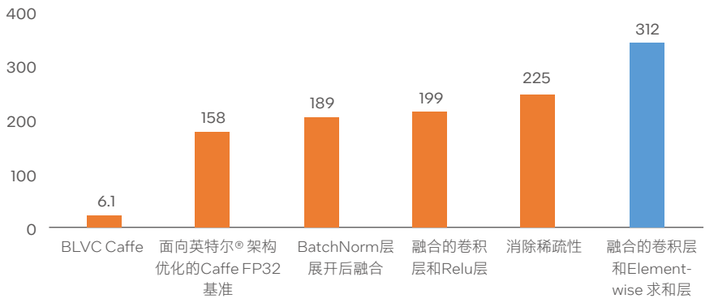


图2-3-6 面向英特尔® 架构优化的Caffe在英特尔® 至强® 可扩展处理器上加入优化方案后，在推理吞吐量和推理时长性能上与BLVC Caffe对比

英特尔® 深度学习加速技术

在第二代英特尔® 至强® 可扩展处理器中，加入了对 INT8 有着良好优化支持的英特尔® 深度学习加速技术，它能够在不影响预测准确率的情况下加速多种深度学习模型在使用 INT8 时的推理速度，有效提升用户深度学习应用的工作效能。

在图像分类、目标检测等深度学习场景中，采用 INT8 等较低精度的数值替代 FP32 是一种良好的性能优化方案。低精度数值可以更好地使用高速缓存，增加内存数据传输效率，减少带宽瓶颈，且在充分利用计算和存储资源的同时，还能有效降低系统功率。另外，在同样的资源支持下，INT8 还可为深度学习的推理带来更多的每秒操作数（Operations Per Second，OPS）。

英特尔® 深度学习加速技术通过 VNNI 指令集，提供了多条全新的 FMA 内核指令，用于支持 8 位或 16 位低精度数值相乘，这对于需要执行大量矩阵乘法的深度学习计算而言尤为重要。

它使用户在执行 INT8 推理时，对系统内存的要求最大可减少 75%[28]，而对内存和所需带宽的减少，也加快了低数值精度运算的速度，从而使系统整体性能获得大幅提升。

* 更多有关英特尔® 至强® 可扩展处理器以及英特尔® 深度学习加速技术的技术细节，请参阅本手册技术篇相关介绍。

利用工具进行模型准确率优化的方法

■ 相似性度量工具

在深度学习中，可以使用相似性度量（Similarity）工具来判断两个特征值之间的相似度。不同的工具可以从不同维度来进行相似性度量，比较常见的有以下几种：

- **欧氏距离**（Euclidean Distance）：是最常见的距离度量，通过对坐标系中的两个点来计算两点之间的绝对距离，距离越大，相似度越低。
- **向量空间余弦相似度**（Cosine Similarity）：使用向量空间中两个向量夹角的余弦值，来衡量两个个体间的差异。与距离度量相比，余弦相似度更加注重两个向量在方向上的差异，夹角越小，相似度越高。
- **标准化欧氏距离**（Standardized Euclidean Distance）：是欧氏距离改进版，在计算各个特征的距离之前，需要先将各个分量进行标准化计算。
- **马氏距离**（Mahalanobis Distance）：用来表示点与一个分布之间的距离，简单而言，单一样本和哪个样本集距离最近，就属于该样本集。

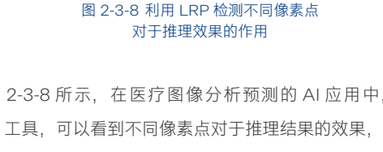


图 2-3-7 利用相似性度量工具分析预测失败原因

利用相似性度量工具，可以灵活地设计和组合出一系列提升模型训练准确率的方法。例如，通过计算两个特征之间的欧氏距离，来分析预测失败的原因。如图 2-3-7 所示，通过测量假阳性样本在特征抽取层和哪个阳性标注最为接近，可以推导出导致误判的主要原因。

■ 层级相关性传播工具

传统上，深度学习模型各层之间的信息传递和逻辑，一直像一个黑盒一样难以回溯，利用层级相关性传播（Layer-wise Relevance Propagation，LRP）工具可以在一定程度上帮助用户解决这一困惑。LRP 工具是利用计算相关性，将相关性逐层向后传播，具有较好的回溯性。同时，利用这一机制，系统也可以推导出哪些因素对预测结果起到的作用更大，从而提升模型准确率。

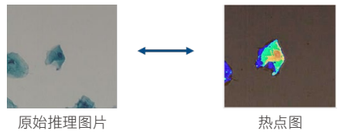


图 2-3-8 利用 LRP 检测不同像素点对于推理效果的作用

如图 2-3-8 所示，在医疗图像分析预测的 AI 应用中，利用 LRP 工具，可以看到不同像素点对于推理结果的效果，并形成热力图，从而帮助方案推导出哪个像素点对最终的预测结果起的作用更大。

江丰生物利用 AI 技术提升高危病筛查效率

背景

当高危病在早期被发现并有效预防，就能帮助病患尽早确诊及早治疗，挽救病患生命。江丰生物与英特尔一起，开始利用先进的 AI 技术，构建和优化基于病理切片的高危病筛查 AI 解决方案，致力于推动高危病的有效防范与治疗。

目前，有几个因素制约着方案的筛查效率和准确率，使其无法进一步提高。首先是数据标注问题：与其他的医疗数据相比，病理切片的分析数据有其独特之处。如图 2-3-9 所示，病理切片图片会有 1 到 40 倍的不同缩放尺度，缩放尺度较小时，图片基本无法进行标注，而当把图片放大到 20 倍甚至 40 倍时候，只能对整张图片中的很小一部分区域进行人工标注，无法覆盖该切片中的所有问题细胞。

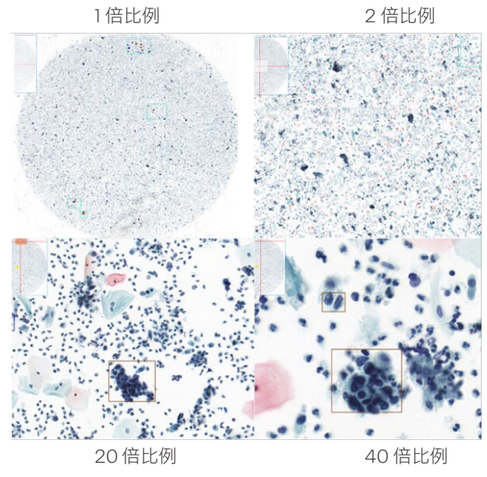


图 2-3-9 不同尺寸的病理切片

[28] 数据源引自 https://software.intel.com/en-us/articles/lower-numerical-precision-deep-learning-inference-and-training

此外，在标注过程中，也存在着标注不完整的问题。有时，标注人员只会标注视野中最严重的问题细胞。如图 2-3-10 上方所示，右下角蓝框中的严重病变细胞被标注了出来，但未标注左上角的红框中的弱阳性细胞；而图 2-3-10 下方，则出现了标注位置不够精准的情况。

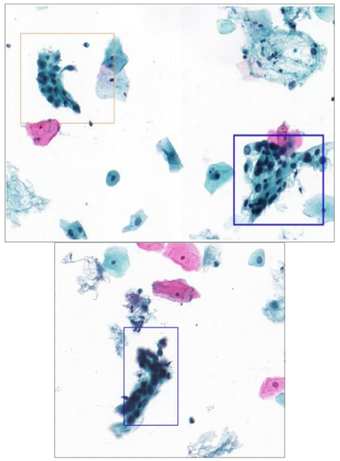


图 2-3-10 标注不够完整的病理切片图片

同时，在目前的标注方案中，通常只关注阳性细胞，对于阴性细胞不够重视。即便对阴性细胞进行标注，也只覆盖到切片级别。对于占总量大多数的阴性细胞，没有有效的利用方案。另外，现有的标注样本严重不均衡，不利于学习效率的提高。

另一个需要关注的问题是神经网络的选择。从实践的效果来看，目前常用的细胞病变目标侦测网络可以输出病变细胞所在位置矩形坐标以及病变细胞具体的描述性（The Bethesda System，TBS）分级，但单独的目标侦测网络并不能很好地解决标注完整性问题。为解决以上这些问题，江丰生物与英特尔一起，从以下几个维度展开优化，以提升筛查深度学习模型的工作效率：

- 优化数据清理和预处理流程；
- 构建两阶段端到端神经网络；
- 引入模型准确率优化工具。

方案与成效

江丰生物联合英特尔构建的基于切片的病变筛查 AI 解决方案，主要工作流程如图 2-3-11 所示，系统在输入图片后，经由数据预处理、分类卷积神经网络和后处理阶段，分别得到阳性预测和阴性预测。对于阳性预测，方案则进行第二阶段的目标侦测网络（基于 ResNet50）模型的训练，然后进行阳性识别的推理过程，并交由医生做最终审查。

图 2-3-11 优化后的方案流程

在优化数据清理和预处理流程中，针对切片图像的不同缩放尺度问题，方案将切片缩放尺度较大、且阳性标注为细胞 / 细胞块级的病理切片图像，采用从大切片图像上裁剪小图的方式来得到训练数据。而针对切片中样本不均衡的问题，训练集采用了阳性：阴性 =1:5 这一比例，同时，由于阳性标注样本相对较少，方案也对样本进行了旋转，以扩大样本的多样性。

同时，为了提升阴性细胞样本的利用效率，方案假设阴性切片中所有细胞均为阴性细胞，阴性切片的训练集从每一张阴性切片上按比例随机裁剪（目的是除去切片边缘干扰）。而对阳性切片的训练集，则直接根据在阳性切片上标注的坐标中心点，加上合理的随机偏移量裁剪为 512*512 的子图。

为提升识别准确率和效率，方案创新地构建了两阶段端到端神经网络。其中，阶段一为分类卷积神经网络，阶段二为目标侦测神经网络。如图 2-3-12 所示，分类卷积神经网络的主要作用是在每张切片产生的滑动窗上进行二分类推理，并对该切片所有的滑动窗结果进行融合处理，从而得到切片级推理结果。

目标侦测网络则是用于对上一阶段确定为阳性的切片进行进一步的阳性区域侦测。

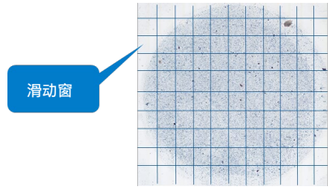


图 2-3-12 基于滑动窗操作的分类卷积神经网络

在模型训练的过程中，方案采用了以下的优化方案来提升训练效果：

- 模型采用了在 Imagenet 数据集上具备优异性能的 ResNet50 来进行训练；
- 训练集准备好后会对其进行旋转，然后按中心点裁剪到 224*224 做均值（Normalize）和归一化（Scale）处理，接下来开始模型训练；
- 鉴于训练集中的正负样本数量较为悬殊，方案将训练好的部分阴性切片和部分阳性切片的子图做集合，递增地加入到训练集中，形成迭代训练。训练集阳性：阴性比为 1:5，从而进一步提升模型的准确率；
- 方案中也加入了相似性度量（Similarity）工具和层级相关性传播（LRP）工具来提升模型准确率。

江丰生物和英特尔一同测评了优化后的基于切片的病变筛查 AI 解决方案，基于 5,961 张精准标注样本进行了训练，并在 246 张测试集上评估了不同的模型。

评估结果表明，加入分类网络后的优化方案，其准确性比单独的目标侦测网络方案有了大幅提升。如图 2-3-13 所示，可以看出，加入分类网络后，当其敏感度（真阳性率，TPR）为 96% 时，特异度（真阴性率，TNR）接近 70%；而在单独目标侦测网络方案中，特异度仅为 40% 左右[29]，这意味着准确性获得了大幅度的提升[30]。

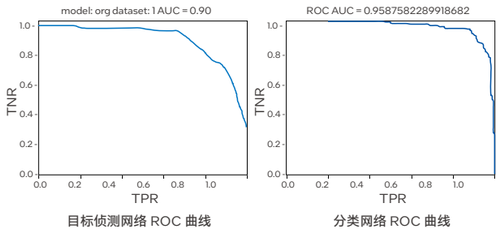


图 2-3-13 优化方案与传统方案准确性对比

江丰生物以 AI 技术助力肺部疾病筛查

背景

随着近年来 AI 技术在医学领域的应用取得飞速发展，基于深度学习 / 机器学习方法的智能化病理分析和诊断技术、正被逐步用于肺部疾病筛查中。

作为一家专业从事数字病理系统开发和生产的高科技生物信息技术企业，江丰生物正致力于以高精度数字化病理切片扫描仪代替传统的显微镜，实现对传统病理切片的数字化转换，并利用基于 AI 的医疗影像处理技术推进智能化病理分析和诊断。现在，针对制约肺部疾病筛查与诊治中的一系列问题，江丰生物正通过筛查系统来推动新型智能化检测技术在该领域的应用。

基于深度学习方法的肺部疾病筛查系统

江丰生物肺部疾病筛查系统，旨在将目标病菌涂片转变为切片数字图像，以便于图像信息的保存和传输，同时在此基础上开发目标病菌相关筛查功能，帮助医生大幅提高判读效率，且解决目标病菌涂片分级的客观性、易控性和重复性问题。

筛查系统基本工作流程如图 2-3-14 所示，首先会应用荧光扫描仪和标注服务平台，对数以千计的目标病菌涂片进行扫描，然后在扫描文件上对目标病菌进行标注。其后再基于深度神经网络进行深度学习，使模型精确识别出目标病菌，以及背景细菌 / 杂质的语义特征。

图 2-3-14 筛查系统基本流程

[29] 该数据援引自江丰生物与英特尔发布的《基于深度学习的病理图像分析》

[30] 数据所使用的测试配置为：双路英特尔® 至强® 铂金 8280 处理器，2.70GHz；核心 / 线程：28/56；HT：ON；Turbo：ON；内存：192GB DDR4 2933；硬盘：英特尔® 固态盘 SC2KG48；网络适配器：英特尔® 以太网网络适配器 X722 for 10GBASE-T；BIOS：SE5C620.86B.02.01.0003.020220190234；操作系统：CentOS Linux 7.6；Linux 内核：3.10.0-957.el7.x86_64；编译器版本：ICC 18.0.1 20171018；Caffe 版本：面向英特尔® 架构优化的 Caffe 1.1.0；工作负载：ResNet50 with 2 classes，130 张图像每秒。

为使系统达到医疗机构应用所需的高效、可靠以及高可用的要求，江丰生物对系统做了如下性能设计：

- **单片识别速度**：基于通用 PC 硬件，可达到单例在 180 秒内完成所有指标识别；
- **目标病菌检测**：目标病菌检测精准率 AP@[IOU=0.5] 大于 80%；
- **痰涂片阴阳性定量分级**：分级准确率（1+ 内）达到 85% 以上。

为达成以上目标，江丰生物将病理学与先进的深度学习 / 机器学习方法相结合，并如图 2-3-15 所示，制定了以下的技术路线设定：

- 在训练阶段，经由涂片扫描数字化、数据标注与数据增强、前景检测模型等步骤，对目标病菌分类器模型（典型的例如 ResNet50）实施训练；
- 在应用阶段，首先通过高性能数字切片扫描仪，得到目标病菌涂片的数字图像，然后采用滑窗法，提取用于深度学习推理的图像 Patch。在获得 Patch 推理结果后，再通过非极大值抑制（Non Maximum Suppression，NMS）算法，剔除重复识别及识别置信度低的检测目标，最终保留高精度的单视野内检测结果；
- 重复以上应用阶段的推理和 NMS 计算过程，最终生成全视野识别的可视化结果与指标，并以此作为辅助筛查系统的输入，为医生显示病历信息、数字图像、目标病菌位置 / 数量以及涂片分级结果等信息，助力其快速筛查诊断病情。

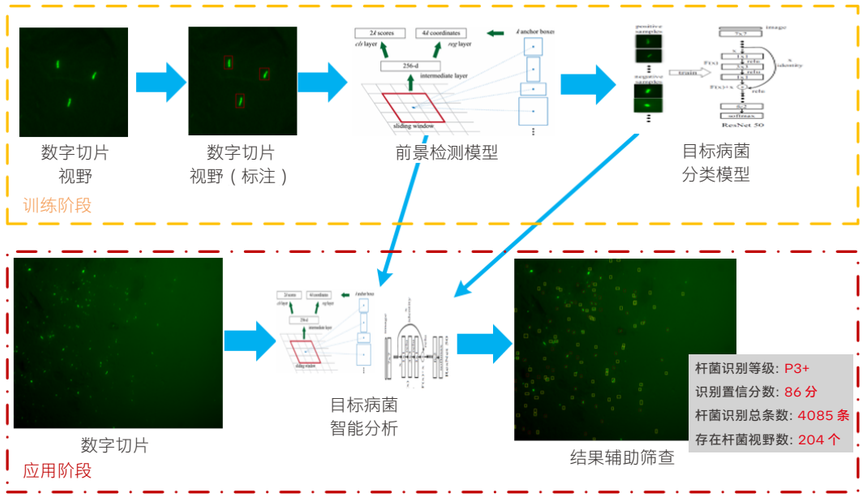


图 2-3-15 目标病菌辅助筛查技术路线图

[31]、[32] 该数据援引自江丰生物内部数据统计。

可以看到，与传统计算机视觉方法相比，上述基于深度学习方法的新方案有着检测精度高，形态适应性强，模型更具鲁棒性等优势。

基于英特尔技术的优化方案与成效

江丰生物在实践部署中发现，医疗机构既有的信息化系统通常都基于 x86 服务器，尤其是基于英特尔® 架构服务器构建。为了帮助医疗机构最大程度地在既有信息化系统上获得更优的处理效能，并有效降低成本，江丰生物与英特尔展开深度合作，在英特尔® 架构平台上对算法模型实施优化，获得更佳的推理速度。

新的优化方案基于 PyTorch 深度模型框架自带的 profile 模块，对模型的各个模块、kernel 运行时间，以及处理器资源占用率等指标进行了全面评估，并采取以下优化措施：

- **PyTorch 优化**：优化前使用的 PyTorch 版本为 1.4，新方案升级到 1.6 版本，其对 native_batch_norm 进行了优化，此项优化经评估可获得约 22% 的 FPS 性能提升；[31]
- **内存管理优化**：考虑到系统内各框架频繁的申请 / 释放内存过程会消耗大量资源和时间，于是新方案引入 jemalloc 用于动态管理优化内存的分配，此项优化经评估可获得约 18% 的 FPS 性能提升；[32]
- **多实例异步处理**：英特尔® 架构处理器不仅具有多核特性，还对大内存有着良好支持，新方案采用多实例异步并发进行处理，能充分利用多核大内存平台带来的优势，以使用 20 个实例进行处理为例，此项优化经评估可获得约 500% 的 FPS 性能提升；[33]
- **整体流程优化**：基于上述优化点，新方案还引入了多实例处理，采用数据加载 DataLoader，对数据输入进行优化，去除冗余部分等方法，使系统的最终工作速度得到了充分优化。

为了验证优化方案在实践部署中的性能表现，江丰生物与英特尔一起，对优化方案进行了测评，测评结果如图 2-3-16 所示。经过各方面优化的方案，性能表现是未优化方案的 11.4 倍。[34]

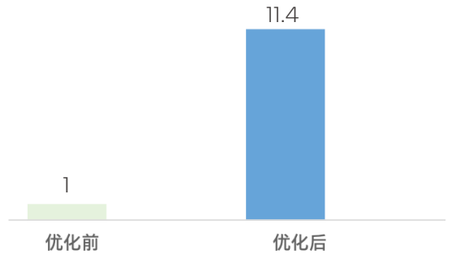


图 2-3-16 方案优化前后性能归一化对比

得益于英特尔® 架构处理器的优异性能以及针对性的优化方案，江丰生物筛查系统已在诸多医疗机构获得了广泛的部署与应用。来自一线的数据反馈表明，新方案能够保持 86.8% 的精准率 AP，以及 88.9% 的涂片级分级准确率[35]，并满足在 80 秒[36] 内对目标病菌涂片完成数字化扫描和涂片定量分级，获得了医院、医生和患者的一致好评。

小结

利用深度学习的方法来对病理切片图像等做出快速检测，不仅可以大大提升医疗机构病理检测的生产力，消弭因专业病理科医生不足带来的一系列问题，也能为病患带去更精确、更及时的治疗方案。目前，基于图像分类和目标检测的病理切片检测 AI 应用，已在众多医疗机构进行了落地部署，并获得良好的反馈。

英特尔® 架构处理器、面向英特尔® 架构优化的 Caffe、英特尔® 深度学习加速技术等在内的一系列英特尔先进产品和技术，已在众多应用场景中，助力基于深度学习的病理切片检测应用大幅提升其工作效率。例如英特尔® 架构处理器对大内存的良好支持，使得在模型训练中可以设定更大的 Batch Size，从而大幅提升训练效率；再如面向英特尔® 架构优化的 Caffe，以及英特尔® 深度学习加速技术对 INT8 的良好支持，可以有效提升推理效率，提升病理切片分析的实时性。

随着英特尔® 至强® 可扩展处理器持续迭代以及其他英特尔新产品、新技术的到来，用户可以基于这些更新的软硬件，来构建训练和推理性能更为强大的 AI 应用。同时，英特尔还计划针对更多的深度学习模型开展推理优化研究，以帮助更多的病患赢得宝贵的治疗时间和效率。

[33] 该数据援引自江丰生物内部数据统计。

[34] 测试工作负载：Medical Image detection, detectron2（detectron2 0.1.1），平台：Dell PowerEdge R740；处理器：双路英特尔® 至强® 金牌 6252 处理器，2.10GHz；核心 / 线程：24/48; 超线程开启；睿频开启；内存 192GB DDR4（12 x 16384 MB 2666 MT/s）；存储：1x 英特尔® 1.8T SSD（英特尔® SSDSC2KB01）；网络适配器：英特尔® C621 (1 x 英特尔® X722 for 10GBASE-T)；操作系统：Ubuntu 18.04.4 LTS（Kernel: 5.3.0-51-generic）；深度学习框架：PyTorch 1.4；库：英特尔® MKL-DNN v0.21.1；实例数：1；优化方案：处理器：双路英特尔® 至强® 金牌 6252 处理器，2.10GHz；核心 / 线程：24/48; 超线程开启；睿频开启；内存 192GB DDR4（12 x 16384 MB 2666 MT/s）；存储：1x 英特尔® 1.8T SSD（英特尔® SSDSC2KB01）；网络适配器：英特尔® C621 (1 x 英特尔® X722 for 10GBASE-T)；操作系统：Ubuntu 18.04.4 LTS（Kernel: 5.3.0-51-generic）；深度学习框架：PyTorch 1.6；库：英特尔® MKL-DNN DNNL v1.2.0；实例数：24。

[35] 该数据援引自江丰生物内部数据统计。

[36] 工作站配置：主板：X11DPI-N，CPU：英特尔® 至强® 金牌 6240R 处理器（24Core，2.4GHZ），内存：192GB DDR4（12 x 16GB，2666MT/S），Raid 卡：LSI 9361-8I，存储：2x Intel 960G SSD，4x 4T SATA 3.5 寸

AI 技术助力药物研发

深度学习方法加速药物筛选

基于 HCS 的表型分类

越来越多的新技术正被运用于加速药物研发进程。基于细胞图像的高内涵筛选（High Content Screening，HCS）方法是目前在系统生物学和药物研发领域常用的自动化分析方法之一，也是 AI 技术在药物发现早期环节的重要应用。其通过显微成像法获得的图像信息，来分析和获得由遗传或化学处理诱导的细胞表型特征。

在这一流程中，对细胞图像的表型检测、分析和分类是最重要的几个环节。但生物学分析过程的固有复杂性和细胞测定的固有可变性，对细胞图像中的表型分析带来了严峻挑战。传统细胞表型特征提取的图像分析方法主要由一系列独立的数据分析步骤组成。如图 2-4-1 所示，在输入原始图像后，首先利用目标检测（Object Detection）方法，在细胞层级或图像层级上提取特征，随后对这些特性进行转换（选择、标准化等），最后是总结归纳相关特征，并作为预测表型的分类算法的输入。

尽管以上的特征检测、分析和分类方法已经在大量药物研发过程中获得成功应用，但其仍存在许多局限性。例如对于对象分割、降维和表型分类，通常需要大量的先验知识，例如所预期的表型几何形态（The geometric properties of the expected phenotypes）要对每个测定流程进行定制。同时，采用传统的 HCS 方法，执行每一个步骤，都涉及方法的定制以及参数的调整。而在对整个分析流程的性能调优过程中，如何对所有参数进行联合优化，以达到性能最优化，目前仍面临挑战，因此整体效率还有待提高。为此，更多基于深度学习的 AI 方法正逐渐被引入基于细胞图像的 HCS 表型分类工作。

基于深度学习的 HCS 方法[37]

背景

在传统的 HCS 图像分析方法中，会将图像数据转换为不同的抽象级别，例如像素亮度（Pixel Intensity）等。在深度神经网络等深度学习方法中，可以通过一个框架来对这些图像数据中的分层抽象进行计算和分析，但这些方法在很大程度依赖手动定义的特征。与之相比，CNN 能够自动地从图像中学习和提取特征，因此在对细胞图像的表型预测中具有更好的效率。

CNN 网络通常包括了输入层、卷积层、ReLU 层、池化层、全连接层等。其中卷积层通过计算层输入（例如原始图像或前一卷积层的输出）和多个二维卷积核之间的卷积，来获得图像中的二维几何信息。每个卷积核都可编码一个几何特征（Geometric Pattern），并可卷积得到一个卷积核映射（或特征映射），该映射是一个基于像素的非线性激活函数，并会被传递到后续的卷积层，获得更复杂的模式。最后，卷积层的输出被送至全连接层，并以前反馈的方式对给定的输入生成预测。

假设 CNN 的输出层有 N_p 个待分类的表型，那么对于给定的输入图像 x，网络将在输出层为计算每一路 j 单元的激活函数 $a_j(x)$，并基于此计算一个向量 ρ，ρ_k 可以构成一个概率质量函数，用于覆盖 N_p 个待分类的表型：

$$\rho_k := p(y = k|\mathbf{x}) = \frac{\exp(a_k(\mathbf{x}))}{\sum_j^{N_p} \exp(a_j(\mathbf{x}))}$$

其中，k 为表型的序号，根据这些概率，可以得到表型的预测值为：

$$\hat{y} = \underset{k}{\operatorname{argmax}}\ p(y = k|\mathbf{x})$$

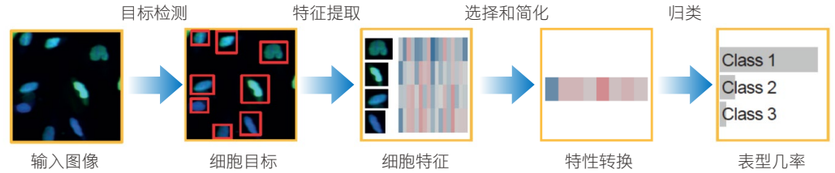


图 2-4-1 传统的 HCS 方法

[37] 本节中有关基于 CNN 及 M-CNN 的 HCS 的技术描述，详情请参阅：Godinez et al, A multi-scale convolutional neural network for phenotyping high-content cellular images. Bioinformatics, 2017

由此可知，诸如层数、卷积层内单元数量，以及卷积核和池化因子的大小选择，都会对预测性能带来影响。而在细胞表型分类中，存在着另外一个问题，即由于细胞本身大小不同，显微成像大小不同，导致在图像数据中往往存在着较大的空间差异，此时如果仍沿用经典的 CNN 网络结构，可能会造成准确率的下降。

多尺度卷积神经网络（Multi-scale Convolutional Neural Networks，M-CNN）可以较好地解决这一问题。与经典 CNN 网络结构相比，其加入了并行的多尺度分析，对于不同尺度上的图像，可以使用不同的 CNN 网络，以独立的方法进行训练。

图 2-4-2 展示了一种具有 7 个尺度的 M-CNN 网络结构，缩放尺寸自上而下逐渐变化。网络在其输入层将输入图像的七个不同尺度的缩放版本，并使用三个卷积层的序列，处理每一个尺度的缩放图像。每个尺度的卷积路径均独立于其他尺度，而在每个尺度的最后一层，都通过汇集方法将得到的卷积核映射缩放到最粗的尺度，并链接起来，用作最终卷积层的输入，最终的输出层将会输出每个表型的生成概率值。

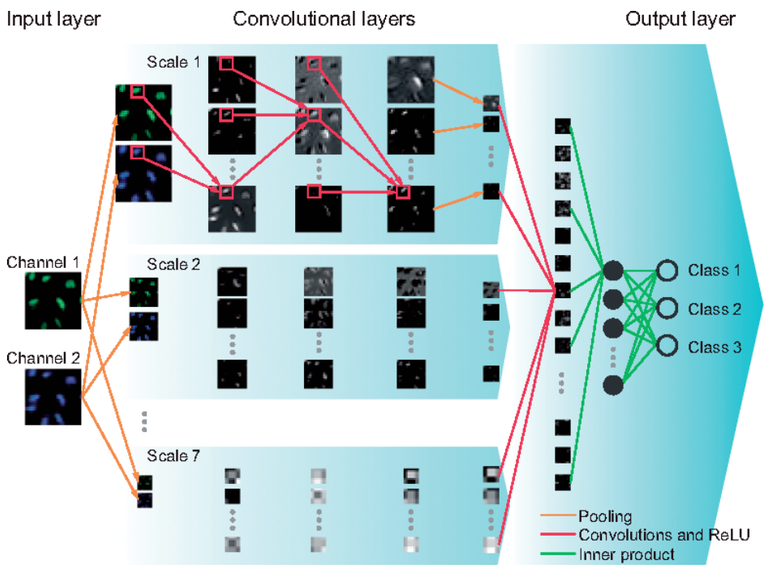


图 2-4-2 M-CNN 架构示意图

软硬件配置建议

对于利用AI技术来加速药物研发，可以参考以下基于英特尔®架构平台的软硬件配置，来进行系统部署。

名称	规格
处理器	英特尔® 至强® 金牌 6240 处理器或更高
超线程	ON
睿频加速	ON
内存	16GB DDR4 2666MHz* 12 及以上
存储	英特尔® 固态盘 D5 P4320系列及以上
操作系统	CentOS Linux 7.6或最新版本
Linux 核心	3.10.0 或最新版本
编译器	GCC 4.8.5 或最新版本
TensorFlow 版本	面向英特尔® 架构优化的 TensorFlow v1.7.0 或最新版本
Horovod	0.12.1 或最新版本
OpenMPI	3.0.0 或最新版本

基于英特尔® 至强® 可扩展处理器的优化

提升单计算节点训练效率

一款新药的研发时间往往长达数年，而其背后常常伴随着患者焦急的等待。为了进一步提升基于 M-CNN 网络模型的 HCS 方法在药物发现工作中的效率，进而让研发得以加速，已经推出了一系列针对英特尔® 至强 ® 可扩展处理器的优化方案，其包括提升单计算节点吞吐量、提升多计算节点效率等多种方法。

首先，在单计算节点上启动 M-CNN 模型进行训练代码如下：

```
python tf_cnn_benchmarks.py
--model=mcnn
--batch_size=32
--data_format=NCHW
--data_dir=INPUT_DATA_DIR
--data_name=mcnn
--num_intra_threads=40
--num_inter_threads=2
--num_batches=2000
--num_warmup_batches=70
--display_every=5
--momentum=0.9
--weight_decay=0.00005
--optimizer=momentum
--resize_method=bilinear
--distortions=False
--sync_on_finish=True
--device=cpu
--mkl=True
--kmp_affinity=="granularity=fine,compact,1,0"
--variable_update=horovod
--local_parameter_device=cpu
--kmp_blocktime=1
--train_dir=TRAIN_DATAWRITE_DIR
```

在单计算节点上，M-CNN 方法遇到的问题之一是内存容量问题。通常而言，深度学习网络的效率可以随着 Batch Size 的增加而有一定程度的提高。用于高内涵筛选的细胞图像通常有着较大尺寸，再加上多尺度联合操作，当 Batch Size 增加到一定量后，所需的内存容量会很大，如图 2-4-3 所示，当 Batch Size 为 32 时，系统所需内存达到了 47.5GB。

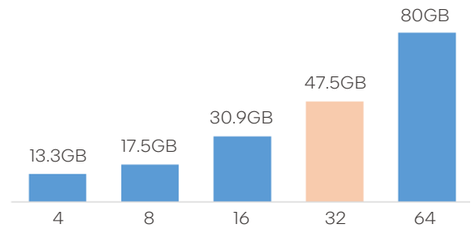


图 2-4-3 不同 Batch Size 下的内存需求量

英特尔® 至强 ® 可扩展处理器对大内存有良好的支持能力，可以有效解决随 Batch Size 增加而带来的大内存需求，其更优化的微架构、更多的核心数量以及对更快、更大容量内存的控制和调度能力，使基于 TensorFlow 框架构建的 M-CNN 方法得以轻松展开。在一项使用 Broad Bioimage Benchmark Collection 021（BBBC-021）数据集[38] 所做的测试中，输入的显微镜图像尺寸为 1024*1280*3，在 Batch Size 为 32 时，单一 TensorFlow 工作进程（Worker）下，处理速度达到 13 张每秒。但这一处理速度对于多达成千上万张图像的数据集而言，整个训练过程仍显漫长，效率亟待提高。

通过NUMA技术的引入，以及基于分布式深度学习框架Horovod的权重同步技术，可以让用户在TensorFlow框架下，同时使用四个TensorFlow工作进程。如图2-4-4所示，在一个典型的计算节点中部署的双路英特尔® 至强® 可扩展处理器，可以被划分为4个计算区域，每个区域分别执行一个TensorFlow工作进程。

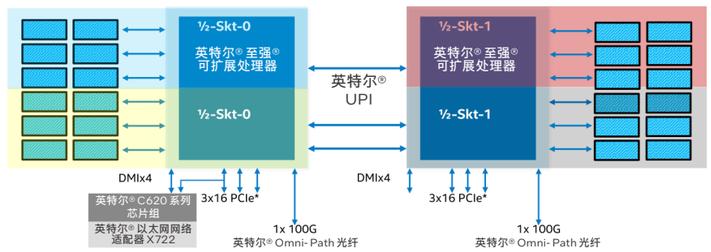


图 2-4-4 典型的计算节点中双路英特尔® 至强® 可扩展处理器的划分

[38] BBBC-021: Ljosa V, Sokolnicki KL, Carpenter AE, Annotated high-throughput microscopy image sets for validation, Nature Methods, 2012

利用 NUMA 的技术特性，可以绑定处理器的不同核心以及不同内存来执行训练，而互相之间不会有计算资源和存储资源的竞争。各个计算区域之间，使用英特尔® 超级通道互联（Intel® Ultra Path Interconnect，英特尔® UPI）技术实现权重同步。通过这种方式，训练模型的吞吐量可获得进一步的提升。如图 2-4-5 所示，使用四个 TensorFlow 工作进程后，在同样 Batch Size 为 32 时，处理速度达到 16.3 张每秒，效率提升达 25.4%。

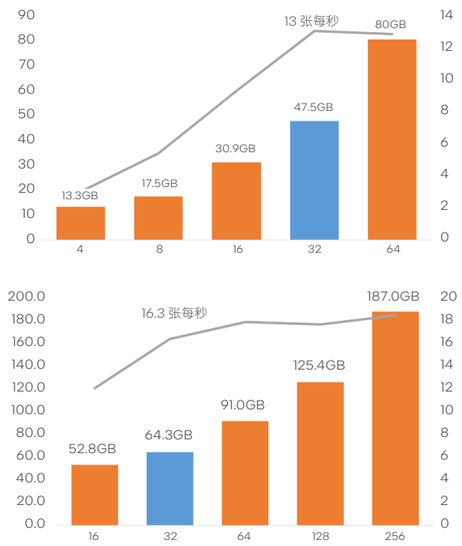


图 2-4-5 TensorFlow 中四个工作线程与单个工作线程性能对比

提升多计算节点训练效率

除了提升单计算节点训练效率之外，利用分布式训练技术方式也可以进一步提升训练效率。在经典的 TensorFlow 分布式架构中，需要使用参数服务器的方法来平均梯度，每个处理线程都可能作为工作线程或参数服务器。前者用于用户处理和训练数据，计算梯度，并把它们传递到参数服务器上进行平均。

但在这一方法中，如果参数服务器的处理能力不足，可能会造成系统的整体性瓶颈。同时，为了实现最优化性能，使用者在一开始就需要指定合适的初始工作线程和参数服务器，但稍有不慎就会带来性能的下降。新的开源 TensorFlow 分布式深度学习框架 Horovod 可以有效解决这一问题。其引入的 Ring-allreduce 算法构建了新的通信策略，允许工作线程来平均梯度，而无需再加入参数服务器。

如图 2-4-6 所示，在 Ring-allreduce 算法中，每个工作线程首先根据各自的训练数据分别进行梯度计算，得到梯度信息。每个工作线程与其他 N-1 个工作线程进行 2*（N-1）次通信。在这一过程中，一个工作线程发送并接收数据缓冲区传来的梯度信息，每次接收的梯度信息被添加到工作进程缓冲区中，并替代上一次的值。所有的工作线程将在发送和接收 N-1 个梯度消息之后，收到计算更新模型所需的梯度。这一方法可以最大化地利用网络能力，避免计算瓶颈出现 [39]。在此通信策略基础上，Horovod 通过开放消息传递接口（Open Message Passing Interface，OpenMPI）建立基于 TensorFlow 的分布式系统。

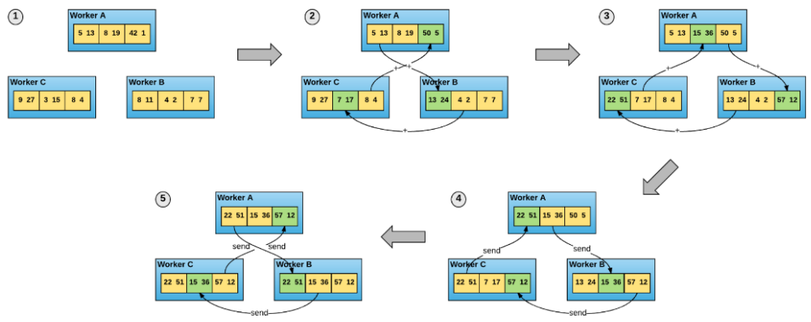


图 2-4-6 Ring-allreduce 算法示意图

即便在采用 Horovod 框架的情况下，所需要传递的梯度信息仍然可观。例如在使用 BBBC-021 数据集所做的测试中，梯度信息大小为 162.2MB。

另一个可以对多计算节点训练效率进行优化的方式是收敛和调整学习率（Learning Rate，LR），不同训练阶段的 LR 大小是深度学习中非常重要的设置项，LR 过大会造成振荡，过小则会收敛速度慢且易过拟合。在基于 TensorFlow 框架构建的 M-CNN 模型训练过程中，可以采用如下的 LR 调整方法来获得性能优化。

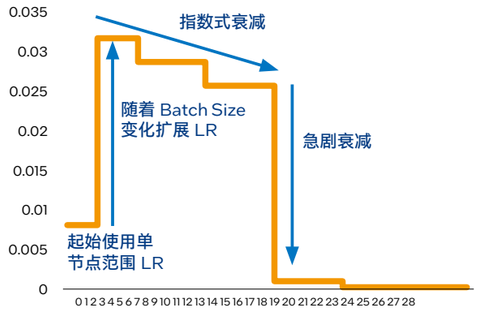


图 2-4-7 M-CNN 网络训练过程中的 LR 调整

如图 2-4-7 所示，在训练之初，首次迭代先使用单节点的 LR，随后将其扩展到全局的 Batch Size 参数。在其后的迭代中，LR 以指数方式衰减，从第 14 次迭代开始，LR 出现一个急剧衰减 [40]。

由此，M-CNN 网络在多计算节点上的训练命令如下：

```
OMP_NUM_THREADS=10 mpirun -np 32 -cpus-per-proc 10
--map-by socket -hostfile HOSTFILE
--report-bindings
--oversubscribe -x LD_LIBRARY_PATH -x
PATH -x OMP_NUM_THREADS -x HOROVOD_FUSION_THRESHOLD numactl -l

python tf_cnn_benchmarks.py
--model=mcnn
--batch_size=8
--data_format=NCHW
--data_dir=INPUT_DATA_DIR
--data_name=mcnn
--num_intra_threads=10
--num_inter_threads=2
--num_batches=2000
--num_warmup_batches=70
--display_every=5
--momentum=0.9
--weight_decay=0.00005
--optimizer=momentum
--resize_method=bilinear
--distortions=False
--sync_on_finish=True
--device=cpu
--mkl=True
--kmp_affinity="granularity=fine,compact,1,0"
--variable_update=horovod
--local_parameter_device=cpu
--kmp_blocktime=1
--horovod_device=cpu
--piecewise_learning_rate_schedule='0.008;2;0.032;5;0.029;10;0.026;15;0.001;20;0.0001'
--train_dir=TRAIN_DATAWRITE_DIR
--save_summaries_steps=1
--summary_verbosity=1
```

[39] 相关技术描述详情，请参阅：Alex Sergeev，Mike Del Balso，Meet Horovod: Uber's Open Source Distributed Deep Learning Framework

[40] 更多 LR 设置技术详情，请参阅：Yang You et el, 2017, "ImageNet Training in Minutes"

诺华利用深度学习提高药物研发效率

背景

作为全球领先的医药企业，诺华正积极借助数字化转型来保持其在药物创新、疾病诊断和药物研究等方面的竞争优势，而“AI+ 药物发现”是其面向未来药物研发进程中的重要一环。

现在，诺华正与英特尔一起，合作研究使用深度学习的方法来加速 HCS 进程。细胞表型的 HCS 是目前诺华进行早期药物发现的重要方法之一。所谓高内涵是指使用经典图像处理技术，从图像中提取的数千个预定义特征（例如大小、形状、纹理等等）的丰富集合。HCS 允许分析显微图像，以研究数千种遗传或化学处理对不同细胞培养物的影响。利用深度学习方法，诺华可以从数据中“自动”学习，并区分一种治疗与另一种治疗的相关图像特征，但细胞显微镜图像巨大的信息量使这一方法仍需耗费大量时间——其图像分析模型的训练时间约为 11 小时 [41]。

现在，英特尔和诺华的生物学家、数据科学家们希望通过基于优化的英特尔® 至强 ® 可扩展处理器上部署的 M-CNN 网络，来加快 HCS 分析。在这项联合工作中，该团队专注于整个显微镜图像，而不是使用单独的流程来首先识别图像中的每个细胞。而且，其使用的数据集 BBBC-021 数据集中的显微镜图像可能比常见深度学习数据集中的图像大得多。

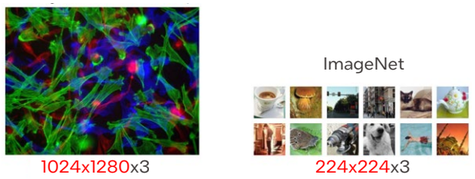


图 2-4-8 用于 HCS 的显微镜图像与常见图像数据集对比

如图 2-4-8 所示，左侧是一个用于 HCS 的显微镜图像，其单张像素接近 400 万，而右侧是来自著名的 ImageNet 数据集 [42] 的图像，其训练数据集单张图像为 15 万像素，双方相差 26 倍。大尺寸的显微镜图像，与其带来的数百万个参数，加之一次训练图像数千个的规模，既对系统内存形成挑战，也带来巨大的计算负荷。为了有效应对这一挑战，双方采用了一系列深度神经网络优化和加速技术，帮助系统能够在更短的时间内处理多个图像，并保持准确率。

优化方案与成效

优化方案在两个方面对基于英特尔® 至强® 可扩展处理器所部署的 M-CNN 模型的训练进行了加速。首先、在单计算节点，充分利用英特尔® 至强® 可扩展处理器对大内存的良好支持，使方案可以采用大 Batch Size（方案中设为 32），并利用 NUMA 技术增加工作线程来提升训练效率；其次、在多计算节点，引入了开源的 TensorFlow 分布式深度学习框架 Horovod，来大幅提升 M-CNN 模型在多节点下的训练效率。同时还设计、采用了优化后的学习率收敛和调整方法来提升性能 [43]。

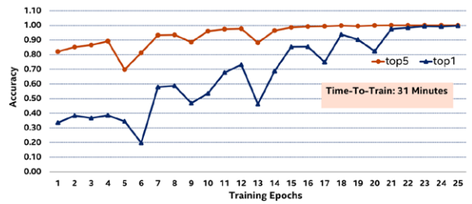


图 2-4-9 诺华优化后方案的训练效果

方案中部署了 8 个基于英特尔® 至强® 可扩展处理器的节点，使用BBBC-021 数据集，图像总量为 1 万张，尺寸为 1024*1280*3。在超过 20 次的训练后，如图 2-4-9 所示，训练时间总长约为 31 分钟，准确率超过 99%。同时，方案在使用 NUMA 技术形成 32 个 TensorFlow 工作进程（每个节点 4 个工作线程）后，处理能力达到了每秒 120 多幅图像，与未优化前相比，性能获得了显著提升。

小结

一款新药从发现、试验到生产，动辄数年，期间伴随着患者及其家属的殷切期待。利用 AI 技术来加速药物研发进程，不仅是众多制药企业加速创新，保持核心竞争力的普遍选择，也是让科技造福人类，助力创造健康生活的重要体现。为此，英特尔也与众多制药企业一起，为加速 AI 方案在药物研发中的应用而努力。

通过合理的优化方案，英特尔® 至强 ® 可扩展处理器在内的先进技术与产品，可以为基于深度学习的 HCS 等 AI 应用提供出色且可靠的大内存支持，以及大 Batch Size 与多 TensorFlow 工作进程支持，来加速单节点或多节点的训练效率，并以高带宽、低延迟的先进互联架构来对 Horovod 分布式训练框架提供支撑，进而大幅加速诺华等药企的药物研发进程。

目前，基于英特尔® 至强® 可扩展平台的一系列 AI 应用，已在众多制药企业获得了落地部署，并获得了良好的效果。值得一提的是，虽然本文中的测试是基于英特尔® 至强® 金牌 6148 处理器展开，但随着第四代英特尔® 至强® 可扩展处理器等英特尔硬件与技术的推出与应用，用户在未来实际部署中可以选用更新的英特尔硬件平台，以及相关软件优化方案来构建性能更强劲的深度学习方案，并获得更佳的训练和推理效果，进而进一步加速药物发现的进程，更好地助力患者治疗与康复。

[41] 该数据援引自 https://newsroom.intel.com/news/using-deep-neural-network-acceleration-image-analysis-drug-discovery/#gs.ptk50k
[42] ImageNet: Russakovsky O et al, ImageNet Large Scale Visual Recognition Challenge, IJCV, 2015
[43] 数据所使用的测试配置为：双路英特尔® 至强® 金牌 6148 处理器，2.40GHz；核心/线程：20/40; HT：ON；Turbo：ON；内存：16GB DDR4 2666*12；硬盘：480GB 英特尔® 固态盘 OS drive*1，1.6TB英特尔® 固态盘data drive*1；网络适配器：英特尔® Omni-Path主机结构接口（HFI）；BIOS：SE5C620.86B.02.01.0008.031920191559；操作系统：CentOS Linux 7.3； gcc版本：6.2；TensorFlow版本：面向英特尔® 架构优化的TensorFlow v1.7.0；Horovod版本：0.12.1；OpenMPI：3.0.0；ToRSwitch：英特尔® Omni-Path架构工作负载：Broad Bioimage Benchmark Collection* 021（BBBC-021）数据集，1万张图像，图像尺寸为1024*1280*3。

AI 助力打造更为精准智能的医疗解决方案

医疗行业中更多 AI 技术的落地应用

更多 AI 方法被应用于医疗行业

近年来，随着不同方向的 AI 技术都获得长足进展，越来越多的 AI 应用也在医疗行业的不同领域获得广泛地落地应用。例如，随着医疗信息化进程的推进，在过去数十年中，医疗数据已逐渐从纸质记录全面走向电子化，这为 AI 方法的应用提供了数据基础。目前，很多医疗机构已经着手部署基于深度学习或机器学习方法的各类 AI 技术，并在医疗科研、临床辅助中取得了良好的成效。

在疾病分析领域，一些医疗机构正尝试运用决策树、随机森林（Random Forest，RF）等机器学习算法，对某种慢性病的海量患者数据进行分析，预测该慢性病患病概率。在数据比对中，基于机器学习的这一慢性病患病率预测方法已被证明可比人工具备更高效率。而另一些医疗机构中，基于海量数据构建的机器学习模型正帮助医师高效评估患者的预后风险得分，从而更好地判断患者临床预后情况，为其选择最佳治疗方案。

除了对已知疾病实施辅助诊疗外，AI 方法还可帮助医疗机构从大量复杂的医疗记录（例如健康信息系统（HIS）中的海量数据）中，利用 NLP 等 AI 技术梳理预测未知的疾病信号，例如从视网膜眼底图像中预测屈光不正等。

本文接下来将就 AI 方法在医疗领域更多的应用方向，包括慢性病预防与诊疗以及放射组学应用等展开描述，并介绍相关的实践案例，探讨 AI 技术在医疗行业中的发展趋势。

AI 方法在医疗领域的重要应用方向

■ 慢性病预防与诊疗

伴随工业化、城镇化带来的生活方式改变，以及人口老龄化进程加速和不健康生活方式的影响，慢性非传染性疾病（以下简称“慢性病”）已成为中国居民的主要死亡原因。一项数据表明，新世纪以来，中国成人慢性病死亡率已占总死亡率的 86% 以上[44]。因此，慢性病业已逐渐成为重大公共卫生问题。

与病原体感染、食物中毒等突发性疾病相比，慢性病具有以下几类特点：

- 慢性病患病人数众多且以中老年人为主，患病率随年龄增长而上升；
- 慢性病多为终身性疾病，治疗护理康复周期长，医疗服务需求量大，护理要求高；
- 大多数慢性病属于不可逆性疾病，不仅影响患者的生活质量，且会给家庭和社会带来沉重经济负担；
- 慢性病往往有交叉并发现象，单一治疗方案难以起效，需多方位综合康复。

基于这样的特点，各级医疗机构对于慢性病的治疗，提出了“预防为主、治疗为辅”的策略，但这需要依据患者健康状况做出综合评估，并长期跟踪。现有专科门诊为主的传统医疗模式及一年一次的体检，显然无法达到早筛查、早发现，给与早治疗。

逐渐丰富和多元化的医疗数据积累，为 AI 技术在慢性病预防和治疗中的应用奠定基础。通过一定算法，机器学习方法可在患者各项健康数据中发现相应的模式，并通过建模学习这些模式，进而对慢性病进行预测。

通过将慢性病预防与诊疗算法部署在医疗机构、康复中心甚至家庭智能设备中，中老年人、肥胖者、烟民等慢性病高风险人群，可以更便捷地得到慢性病风险评估、个性化健康干预以及干预效果长期评估，更好地实现自我健康管理。

■ 放射组学应用

自 2012 年第一次被提出以来[45]，放射组学（Radiomics，亦称影像组学，本文中统一采用放射组学）就一直受到医疗行业的热切关注。其是指从 CT、MRI、PET 等医疗影像中，以（半）高通量方法提取大量影像信息，通过区域分割、特征提取和模型建立等过程，来对影像数据信息进行更深层次的挖掘、预测和分析，从而辅助医生做出更精准的诊断，已在诸多疾病诊断治疗中，发挥着越来越关键的作用。

放射组学融合了基因信息和影像多模态信息，可将影像转换为可挖掘的高通量影像特征数据，量化病灶组织的空间 - 时间异质性，揭示出肉眼无法识别的疾病特征，有效将医学影像转换至高维的可识别特征空间，并使用统计学和 / 或机器学习的方法，筛选最有价值的影像组学特征用以解析临床信息，从而建立具有诊断、预后或预测价值的模型，为精准个体化诊疗提供有价值的信息。与活检方法相比，放射组学分析不仅可以全面提取病例特征，还可以重复利用数据；与传统医学影像相比，

[44] 数据引自国家卫生计生委疾病预防控制局发布的《中国居民营养与慢性病状况报告（2015年）》

[45] 放射组学（Radiomics）由荷兰学者Philippe Lambin在其论文《Radiomics: Extracting more information from medical images using advanced feature analysis》中首次提出：https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4533986/

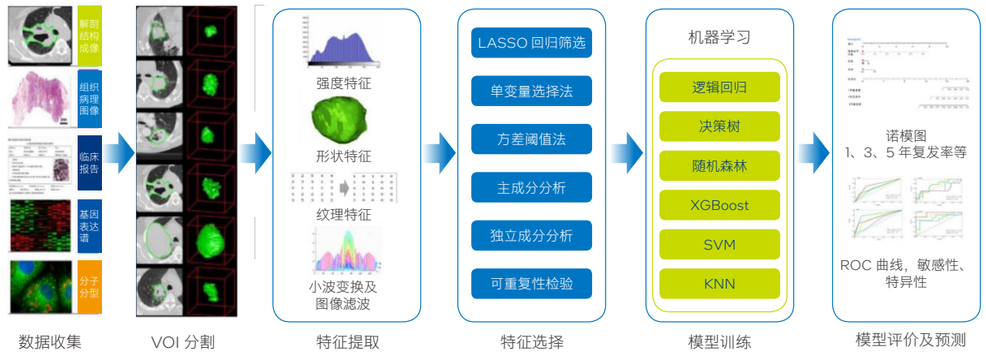


图 2-5-1 放射组学基本分析流程

放射组学具有高通量、定量、计算速度快、精度高等优点，因而得到了研究人员广泛关注与研究。

放射组学基本分析流程如图 2-5-1 所示，分为数据收集、感兴趣容量（Volumes Of Interest，VOI）分割、特征提取、特征选择、模型训练以及模型评价及预测几个主要步骤。在数据收集阶段，系统会将患者的 CT、MRI、PET-CT 等以 DICOM 影像格式导入，基因表达谱以及临床报告则以特定的临床信息格式载入，入组数据需要具有相同或相似的采集参数，保证数据不会受到机型、参数的影响。

考虑到纳入研究的影像数据可能来自不同的扫描参数或扫描机器，为了尽可能减小由此造成的影像数据差异及其对最后结果的影响，平台会对每例影像进行重采样并通过 BSpline 函数进行插值，以便保证后期处理时每组影像的分辨率相同，并进行信号的归一化处理。

VOI 分割是指在影像图像上勾画出感兴趣区域，从而针对这一特定区域计算放射组学特征。特征提取是通过提取强度、形状等特征，将低维视觉特征、高维复杂特征和临床经验特征相结合，来全面分析病灶异质性。

然后，通过最小绝对收缩选择算子（Least Absolute Shrinkage and Selection Operator，LASSO）回归筛选、主成分分析法（Principal Component Analysis，PCA）等特征选择方法，在筛选特定的放射组学特征后，通过逻辑回归，决策树等机器学习方法进行模型训练。最后，系统会通过受试者曲线（Receiver Operating Characteristic Curve，ROC）、诺模图等对模型效果做出评估，并进行预后预测。

在以上流程中，选择合适的特征选择方法将影响整个放射组学系统的预测效率和精度。通常系统需要根据影像采集参数的不同，以及呼吸运动位移带来的干扰等，使用合理的特征选择方法来筛选抗噪声能力强的放射组学特征，并通过调整参数来提高其稳定性。另外，特征选择也是避免“维度灾难（Curse of Dimensionality）”和信息损失的关键环节。

针对医疗机构 IT 基础能力储备不强的现状，信息化厂商在提供放射组学方案时，也会引入数据可视化工具来帮助医生，并通过一键式的操作降低使用门槛。在大量医疗影像病灶经输入后，通过一键提取特征值并进行归一化处理，能迅速给出具有统计学价值的特征值，供机器学习模型进行训练，从而有效提升模型效率和精准度。

现在，基于放射组学的一系列医疗科研、辅助诊疗方案已在众多医疗机构得到了部署和实践，并在病灶的早期筛查等场景中取得了显著的效果。

■ 利用 NLP 技术开展医疗信息整合

存储并流转在各个医疗信息化系统中的各类数据有其独有的语言和文本特征，因此，传统的自动化系统很难全面地利用并分析数据，用以提供患者治疗和管理。尤其是非面向医疗行业的应用系统一般都缺乏有效和系统化的方式，来确定医疗数据结构，以及整合并分析数据和结果，势必也难以帮助医生获得洞察并做出更精准的临床决策。

NLP 技术的发展可以帮助医疗机构更有效地对不同类型的健康数据开展洞察。一般地，基于 NLP 技术构建的系统在工作时，关键的算法模块包括了基于医疗文本的命名实体识别模型以及关系提取模型。

■ 命名实体识别

实体是知识或概念的基本要素。这是一个可以唯一识别并与其他实体区分开的对象。NLP 解决方案中的命名实体识别既可以总结并区分这些在医疗文本中出现的概念，又可以组织概念体系，如患者、患病部位、疾病和症状。

命名实体识别模型由一个神经网络模型和一个经过预先训练的语言模型组成。这些模型使机器能够自动识别医疗文本中出现的实体，包括识别实体的边界以及确定实体的类型。

■ 关系提取

语义关系用以描述实体和概念之间的关联与交互。这些关系是知识的核心组成部分之一。例如，患者和疾病之间存在一种诊断关系，而且疾病有不同的症状。

关系提取模型可自动识别文本中不同实体间的语义关系。该模型可以形成三元组，从而生成一个语义网络，匹配文本，对文本进行结构化处理，并以图形数据的形式存储。

通过 NLP 技术的加入，医疗机构就有能力将不同维度、具有不同特征的医疗数据，包括临床记录、影像报告、实验室测试、探视记录等开展有效整合，揭示非结构化和不相关的医疗数据点中隐含的信息，提供患者数据的整体视图，帮助医生做出更精准的临床决策，并为患者提供更好的治疗方案，也有助于推进临床研究。

■ 利用 OCR 技术加速医疗信息流转

传统的医疗信息采集、录入和转化流程中，相关的住院、用药以及就诊等信息的采集和转化都需要人工参与。不仅耗时耗力，而且还可能因为人为疏忽导致错录、漏录等问题。基于 AI 方法的 OCR 技术的推出，成为解决这一问题的良方。

作为 CV（计算机视觉研究）领域的重要分支，OCR 技术是利用光学和计算机技术将图像中的字符信息读取出来并转化为系统数据。OCR 系统的工作流程一般可分为以下几个步骤：

- **预处理**：对待提取字符信息的图像进行降噪、矫正和加强，包括几何变换（透视、扭曲、旋转等）、畸变校正、去除模糊、图像增强和光线校正等；
- **文本检测**：检测文本的所在位置、范围及其布局，即发现文本所在区域、文本范围有多大。常用的 AI 模型包括 Faster R-CNN、RRPN、DMPNet、CTPN 等；
- **文本识别**：这一步是在文本检测的基础上，对文本内容进行识别，将图像中的文本信息转化为文本信息。常用的 AI 模型组合包括 CNN + RNN + CTC、CNN + RNN + Attention 机制等。

目前在医疗行业中，基于 AI 方法的 OCR 技术正为医疗机构的效能提升带来巨大助力。例如在医疗票据录入场景中，不同使用者（医生、病人、病人家属以及医保机构等）都有可能需要将医疗票据中的文本信息录入系统，以便进行下一阶段的操作。通过智能 OCR 产品的应用，可以将上述手动录入流程转化为自动流程，从而提高信息采集、录入和转化效率和正确率，实现医疗信息管理智能化、精细化。

英特尔® 架构提升机器学习方法效率

医疗领域中的常用机器学习方法

机器学习方法是医疗行业中常用的 AI 技术分支，常见的机器学习方法可以分为分类、回归等不同范畴，以下内容将简要介绍在医疗行业中常用的一些机器学习算法。

■ 决策树与随机森林算法

决策树是一个树形结构的监督学习模型，模型会对每一个特征进行判断，产生不同的结果并进行分支，每个分支再对特征进行判断、继续分支，直到该分支不满足拆分条件为止，最终推断出分类结果。

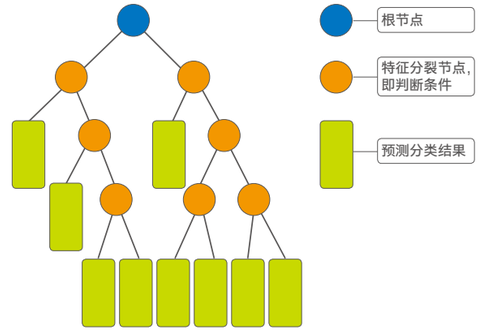


图 2-5-2 决策树模型

[44] 数据引自国家卫生计生委疾病预防控制局发布的《中国居民营养与慢性病状况报告（2015年）》
[45] 放射组学（Radiomics）由荷兰学者Philippe Lambin在其论文《Radiomics: Extracting more information from medical images using advanced feature analysis》中首次提出：https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4533986/

决策树的深度增高容易出现过拟合现象，随机森林算法可以有效解决这一问题。简单而言，随机森林作为一种集成学习方法，其用随机方式构建一个决策树森林，当有一个新的样本进入随机森林，就让每一棵决策树都进行一次判断，计算样本的分类，然后通过“投票”的方式来得到预测样本的归类。随机森林既可用于分类，也可用于回归问题，而这两类问题恰好构成了临床诊断中所需要着重解决的一些定性和定量问题，例如特定人群筛查。

■ 逻辑回归算法

逻辑回归（Logistic Regression，LR）算法是目前常见的机器学习算法之一，其在线性回归的基础上引入 Sigmoid 函数，将线性回归（- ∞，+ ∞）值域映射到（0-1）之间，进而用于预测某种疾病发生的概率。例如在传染病防控模型中，将患病毒感染设定为 a=1，未感染设定为 a=0，将 N 个独立样本中的特征值 b（年龄、性别、病史、旅行史、接触史等）引入如下目标函数中：

$$G(a=1|b;\theta)=\frac{1}{1+e^{-\theta^{T}b}}$$

然后利用最大似然求解极大值，并引入正则项优化，惩罚过大参数避免过拟合，从而计算得出最优的参数值。最终，通过样本数据训练得出是否感染的概率模型。

LR 模型可对连续的数值特征进行离散化，易于模型快速迭代且具有较强的鲁棒性。在对离散后的向量进行特征交叉后，更有助于提升模型表达能力。

■ 几种 Boosting 算法

集成学习是机器学习中通过一系列弱分类器来产生强分类器的方法，当弱分类器间存在强依赖关系，如图 2-5-3 所示，各个弱分类器之间有着串行关系时，称之为 Boosting 算法。典型的 Boosting 算法包括了自适应提升（Adaptive Boosting，AdaBoost）、梯度提升迭代决策树（Gradient Boosting Decision Tree，GBDT）算法。

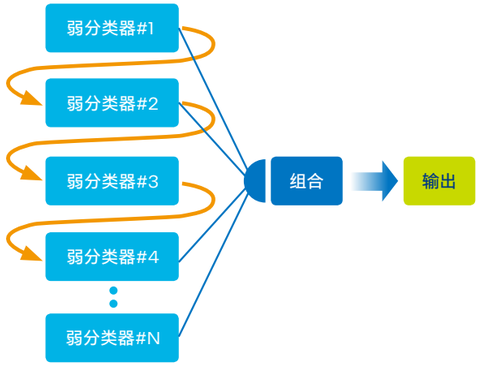


图 2-5-3 机器学习中的集成学习方法

AdaBoost 算法采用的是迭代的思想，一般采用单层决策树作为弱分类器。每次迭代只会训练一个弱分类器，然后让计算好的弱分类器参与下次迭代。N 次迭代后会出现 N-1 个训练好、参数不变的弱分类器，以及第 N 个需要进行训练的迭代器，模型的最终效果取决于 N 个弱分类器的综合效果。在 AdaBoost 算法的训练过程中，每次迭代均会更改样本权重和对应的弱分类器权重，因此其可以根据不同弱分类器的特性不断进行调整。

GBDT 算法是由一系列分类回归树（Classification And Regression Tree，CART）集合而成的强分类器。CART 回归树是在二叉树上不断根据特征进行分叉，例如当前树节点 J 是基于 a 个特征值进行分叉，则特征值小于 b 的样本划分为左子树，大于 y 的样本划分为右子树：

$$J_1(a,b)=\{X|X^{(a)}\le b\}\text{（左子树）}$$

$$J_2(a,b)=\{X|X^{(a)}> b\}\text{（右子树）}$$

CART 回归树实质就是在该特征维度上对样本空间进行划分，典型 CART 回归树产生的目标函数为：

$$\sum_{x_i\in J_m}(y_i-f(x_i))^2$$

与 AdaBoost 算法一样，GBDT 算法也采用了迭代的方法，其目标函数也可以表示为：

$$F(x)=\sum_{i=0}^{M}a_i f_i(x)$$

但与 AdaBoost 算法相比，GBDT 算法每一轮预测和实际值有残差，下一轮再根据残差进行预测，最后将所有预测相加，就得到了预测结果，更重要的是，GBDT 算法具备较强鲁棒性，对于复杂的数据采集尤为重要。

近年来广受关注的 XGBoost 是 GBDT 算法的一个优良扩展和高效实现。其核心思想，就是通过不断进行特征分裂来生成新的分叉树，每添加一个树，其实就是学习一个新函数来拟合上次预测的残差。因此，XGBoost 目标函数可以定义为：

$$y_i=\sum_j q_j x_{ij}$$

当有 k 个样本，其第 N 轮的模型预测结果为：

$$y_i^{(N)}=\sum_{k=1}^{N}f_k(x_i)=y_i^{(N-1)}+f_N(x_i)$$

与 AdaBoost、GBDT 等算法相比，XGBoost 算法有着如下的优势：

- XGBoost 支持并行计算，可充分利用处理器的多线程能力，尤其当其工作在英特尔® 架构平台上时，能更有效利用英特尔® AXV-512 等新指令集提升矢量计算能力；
- XGBoost 在其代价函数中引入了正则化项，可以有效地控制模型的复杂度，防止模型过拟合；
- XGBoost 支持列抽样（column subsampling）方式，不仅能够防止过拟合，还能降低计算复杂度。

■ LASSO 算法

众所周知，当模型在样本特征很多且样本数相对较少时，容易陷入过拟合。缓解过拟合问题一般可以用两种方法。一是减少特征数量，二是通过正则化来减少特征参数 w 的数量级。所谓正则化，即是指选择平均损失函数和模型复杂度同时较小的模型。因此，LASSO 等算法的目标在于对引入的正则化项（表示模型复杂度的单调递增函数）实施优化，正则化项越大，模型复杂度则越低，过拟合概率也就越低。

正则化项可以是模型参数向量的范数，常用项有 L1 范数、L2 范数。LASSO 算法即是对 L1 范数的正则化，其优化目标可表示为：

$$\frac{min}{w}\sum_{i=1}^{m}(y_i-w^{T}x_i)^2+\lambda||w||_1$$

其中正则化参数 λ > 0。同时，L1 范数正则化还有更易获得稀疏解的优势，即其求得的 w 会有更少的非 0 分量。LASSO 算法求解通常可采用近端梯度下降法（Proximal Gradient Descent，PGD）[46]。

而 LASSOCV 是沿着正则化路径迭代拟合的 LASSO 线性模型，其是基于 LASSO 方法，加上 K-Fold 交叉验证来自动找出最优模型。交叉验证是机器学习方法建立模型和验证模型参数时常用的办法，即将数据集切分成多个部分，每个部分轮流作为测试集，以验证在其余数据上训练出来的模型。K-Fold 交叉验证就是将数据集切分为 K 个子集后，进行交叉验证的一种方法。

目前 LASSO 算法已被广泛地应用于压缩感知、图像处理、趋势分析等领域。

* 更多 LASSO 算法内容，可参阅周志华教授所著《机器学习》西瓜书 11.4 节部分内容。

■ 堆栈式集成学习（Stacking）算法

在现实机器学习任务中，数据往往存在特征分层的现象，高层级的特征语义信息复杂，往往难以用单层简单模型提取的信息加以表示，因而无法获得好的预测结果。Stacking 默认通过 2 层模型来实现复杂层次特征的提取，以获得对数据更好的拟合。

在第一层中，可在原始训练集（Xtrain0, Ytrain）上训练不同类型的基模型（level-0），并利用基模型骨干网络从原始验证集（Xvalidate0, Yvalidate）的输入上提取特征，合并组成新的训练集（Xtrain1, Yvalidate），并在原始测试集（Xtest0, Ytest）的输入上提取特征，合并组成新的测试集（Xtest1, Ytest）。

在第二阶段，可利用新的训练集（Xtrain1, Yvalidate）和测试集（Xtest0, Ytest）训练不同类型的模型，融合后作为元模型（level-1）。模型部署时，通过 level-0 推理以提取初级特征，并输入 level-1 以输出最终预测结果。当然，为了获取更复杂的语义特征，Stacking 也可以实现从 level-0 到 level-N 层模型的不断堆叠。

[46] 以上 LASSO 相关算法描述，部分参考周志华教授所著《机器学习》西瓜书 11.4 节部分内容。

■ PCA 算法

PCA 算法是一种使用广泛的数据降维算法。降维是对高维特征数据进行的预处理，通过去除高维数据中的噪声和次要特征来加快数据处理速度，提升机器学习模型效率。简单而言，如图 2-5-4 所示，PCA 算法是通过将数据坐标轴（蓝色坐标轴）上的基线（红色线）进行旋转，一直旋转到数据方差最大（三角形数据在基线上投影最大）的方向，然后通过特征值分析来确定需要保留的主成分个数，进而实现数据降维。

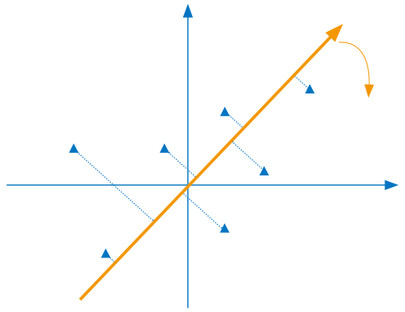


图 2-5-4 PCA 算法映射示意

一般地，假设有 X 行 Y 维原始数据，PCA 算法基本步骤如下：

- 读入数据矩阵，将数据按列组成 Y 行 X 列矩阵 Z；
- 将 Z 的每一行（代表一个属性字段）进行零均值化，即减去这一行的均值；
- 计算协方差矩阵；
- 计算协方差矩阵的特征值及对应的特征向量；
- 将特征向量按对应特征值大小从上到下按行排列成矩阵；
- 保留前 W 行组成矩阵 D；
- 将数据转换到个特征向量构建的新空间，Y=DX 即是降维到 k 维后的数据。

与其他降维算法相比，PCA 算法有着以下优点：

- 属于无监督学习，不受参数限制；
- 各主成分之间正交，可最大程度消除数据成分间的相互影响；
- 计算开销低，易于实现；
- 能有效去除噪声；
- 可达到数据压缩的效果且信息损失小。

基于这些优点，目前 PCA 算法已被广泛运用于高维数据集的探索与可视化，以及数据压缩、医疗 / 金融数据预处理、语音分析等领域。

医疗应用中的高维机器学习模型

根据数据的二八定律，传统专家规则系统可以通过人为总结的经验来覆盖 80% 的人群。然而，二八定律又叫关键少数法则，也就是说在任何一组事务中，最重要的只占其中的 20%，其余 80% 尽管是多数，却是次要的。不过，剩下的 20% 人群若通过规则覆盖，需要的维度会高出几个量级。此时如果通过机器学习模型来对医疗数据进行挖掘，将特征维度提升至百万至亿级别，就可以有效覆盖后 20% 的长尾用户。

具体到医疗应用的具体场景中，传统专家规则系统可能仅通过医学检查结果来判断用户是否确诊，或根据典型症状，例如发烧、起疹子等对疾病进行筛选甄别。而通过机器学习模型，可以通过更多的拓扑关系，例如用户本身的健康记录，是否是某种疾病高危群体等关键信息构成高维组合特征，可以在规则模型的基础上大大提升疾病判断的覆盖面和识别的准确率，在确保提升召回率的同时，还能维持较高的准确率。目前第四范式等企业推出的离散化高维模型，已经可以将维度提升至千万，乃至上亿级别。

通过构建高维机器学习模型，可以带来以下几个方面的优势：

- **特征（规则）带来的高维**：每个特征对应业务上的一条规则，业务规则是人为总结出来的，数量少（一般千条以内），对真实世界描述能力就差；而高维模型所使用的规则（特征）在百万级，远大于一般业务模型，可以大幅提升对预测和识别的准确率；
- **模型（非线性）带来的高维**：包括规则模型在内的线性模型表达能力较弱，且线性模型的非线形化需要基于核函数、手工离散化和特征组合等方法，在学习之前就要付出大量人力工作。而树模型可以通过海量真实数据的输入，产生高度非线性的模型。维度和样本数成指数级关系，对真实世界的表达能力更强；
- **模型融合带来的高维**：虽然高维模型表达能力很强，但无限制地提升维度会导致过拟合。而每个分类器通过高维捕捉数据的不同方面，通过模型融合能刻画更高维度。另外模型融合隐含正则，也可以防止出现过拟合。

英特尔为高维机器学习模型提供更强硬件基础设施支持

高维机器学习模型。不同于一般算法模型，往往构建的是一个巨型的金字塔型数据矩阵，其底层的数据维数可能高达上亿级别，因此其在带来优势的同时，对通用计算能力以及海量内存有着迫切需求，对基础硬件设施的性能要求更高。

基于英特尔® 架构的处理器更高的处理器时钟频率、更多的处理器内核和线程无疑可为高维模型提供更强算力支撑。第二代英特尔® 至强® 可扩展处理器不仅具有多达 56 个处理器内核、112 个线程以及全面升级优化的微架构，还配备了更快、效率更高的高速缓存来提升处理效能，并可支持高达 36TB 的系统级内存容量。其集成的英特尔® AVX- 512，可提供更宽的矢量计算功能，能对机器学习中的多种算法提高执行效率给与有力支撑。

与此同时，高维模型意味着系统必须应对海量的数据处理。通常，当数据的维度在百万级时，文件大小为 GB 级，而在十亿维度时，文件大小可至 TB 级。机器学习系统无论使用何种算法进行模型计算和更新时，都会产生大量的中间结果数据用于模型迭代，这些中间结果的存储性能显然直接制约了训练速度的进一步提高。同时，在一些场景下，还需要中间数据在发生意外时不会丢失。

在传统的基础硬件设施中，高性能的存储需求一般都是由 DRAM 内存来承担。但随着数据的维度到达一定量级，就需要更为经济可靠的硬件设施来提供存储能力。以独特的 3D XPoint™ 存储介质构建、能兼顾高性能和大容量两方面需求的英特尔® 傲腾™ 持久内存显然是良好的选择，其提供了两项高维机器学习模型所需的重要特性：高密度和持久性。前者意味着高达 512 GB/ 每 DIMM 插槽的内存最大密度，是目前 DRAM 内存的数倍，而后者则使得服务器即便发生断电或重启，数据仍可保留。

应用案例

第四范式构建慢性病预防与管理闭环管理方案

■ 背景

慢性病已对人们的生活质量和社会经济造成了巨大的危害，而对抗慢性病最有效的措施是进行有效的预防。如图 2-5-5 所示，慢性病防治可简略为四步法则：

1）为肥胖、吸烟、中老年以及有既往病史者等高风险人群建立健康档案；
2）通过科学的方法进行慢性病风险评估；
3）采取有效的个性化健康干预方案，例如运动方案、饮食方案等；
4）对干预的效果进行长期跟踪，判断风险趋势，调整干预方案。

过去，以上工作都需要经验丰富的专业医师、健康专家、营养专家以及运动专家等给出专业的意见。但在医疗资源日益紧张的今天，为大众提供普遍性的专家服务显然并不现实。此外，即便是专家服务，也是依赖个人经验进行判断，难以满足精准与个性化需求。那么，如何利用高科技手段，为更多居民提供高质量慢性病预防和管理服务，就成为众多医疗健康机构和高科技企业新课题。

基于丰富的医疗数据，通过机器学习的方法来实施风险评估、个性化健康干预以及干预效果评估，已经成为应对慢性病挑战的有效途径。基于这一模式，第四范式与上海交通大学医学院附属瑞金医院共同合作，结合瑞金医院精湛的专业知识和丰富临床经验以及全球最大的代谢性疾病样本库，使用第四范式的

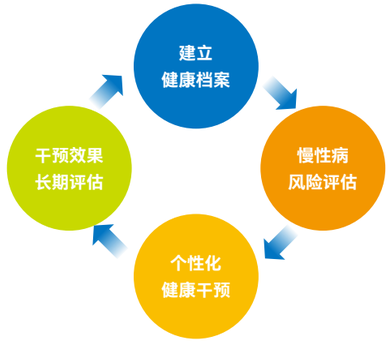


图 2-5-5 慢性病防治四步法

机器学习“先知”平台，利用国际领先的机器学习技术，同时采用英特尔先进软硬件产品，构建了知宁慢性管理系列产品，包括知宁慢病管理一体机、慢病管理云系统、瑞宁知糖、瑞宁知心、慢病管理随访箱、健康小式机器人等产品，助力医疗健康机构实施慢性疾病的全流程预防管理。

■ 方案与成效

如图 2-5-6 所示，第四范式开发的慢性病预防管理服务主要由知宁慢病管理一体机和慢病管理云系统组成，用户可以通过登录慢病管理云系统，借助慢病管理一体机进行智能检测，建立自己的慢性病管理档案。检测数据上传到云平台后，通过精准的机器学习模型对检测数据进行慢性疾病风险精准评估，并结合多病种风险因素分析，提供科学、个性化健康干预方案；同时，方案还利用“健康范式”微信公众号及小程序为用户提供智能提醒及跟踪管理服务，实现干预效果的长期评估管理。通过以上的闭环系统，用户即可获得集体检、筛查、干预和管理于一体的全方位慢病管理服务。

图 2-5-6 慢性病预防管理闭环

目前，知宁慢病管理一体机已能为用户提供血压、血糖、尿酸、胆固醇、血氧、心电图、脂肪率、代谢指数、肌肉含量（%）、水分含量（%）、体温等近 20 项指标检测。云系统采用半监督多任务 SMT-GBDT 机器学习算法，基于全球最大最新代谢性疾病样本库，建立了针对当前中国人的慢性病高精准筛查模型，筛查范围包括多种高发慢性病。从实践来看，模型评估效果远优于现行标准（包括美国、芬兰等发达国家标准及中华医学会标准），预测结果准确率达到专业医生目前使用的临床金标准的 2 到 3 倍 [47]。

先进算法的背后，是第四范式构建的超高维的机器学习方法。从前文对高维模型的描述可知，机器学习中模型的维度越高，学习能力就越强。在传统基于专家经验的慢性病管理中，穷尽医师多年的经验积累，往往也只能总结出数千条专家规则，以此来面对当代人多样化的生活方式和更高要求的慢病管理，显然越来越力不从心。

云系统通过对检测数据进行切片，形成了超高维的机器学习能力来应对这一挑战。在数据预处理环节，系统首先对全量样本进行数据建模。在特征工程阶段，抽取检测结果数据、用户信息等基本信息，结合检测者的历史医疗记录，家族病史等多样化特征，再利用超高维的机器学习算法，以及基于英特尔® 架构处理器的服务器集群所构成的强悍算力，通过对数据原始字段进行超高维组合和衍生，最终形成总量达上千万乃至数以亿计的特征集。

与传统机器学习算法相比，第四范式 GBDT 机器学习算法在模型准确度、离散特征使用能力等多个方面都优于决策树等模型，如表 1 所示，第四范式 GBDT 算法可以兼顾模型准确度要求以及防止模型过拟合的要求，同时在支持的建模样本数量和输入特征数量上，也比传统集成学习决策树算法有着大幅提升。

	传统决策树算法	第四范式 GBDT 算法
树的数量	单棵树	多棵树
模型准确度	树过深容易过拟合，刻画准确和过拟合难以兼得	用很多棵简单的树迭代，不容易过拟合
样本数量	几百万级	上亿级
输入特征	数千	没有限制，由平台节点规模而定
离散特征使用能力	无法处理大规模离散特征	可实现大规模离散特征的处理和使用

表 1 第四范式 GBDT 算法与传统决策树算法比较

为提升慢性病预防管理效能，第四范式在整个闭环的各个流程中，都引入了英特尔® 架构产品来提升效率。一方面，采用第二代英特尔® 至强 ® 可扩展处理器的加入，让平台有了足够算力，来应对万亿级高维数据处理提出的挑战。同时，处理器中所集成的英特尔® AVX- 512 技术，也能以强大的矢量计算能力，加速模型预测过程。另一方面，英特尔® 傲腾™ 固态盘则将高吞吐量、低延迟、高服务质量和高耐用性结合在一起，为平台提供了高质量的数据存储基础设施。

目前，新方案在多家医疗机构的实践中已被证明具有良好的表现。

[47] 数据援引自第四范式《第四范式知宁慢病管理一体机》产品手册。

面向英特尔® 架构优化的 Python 分发包，助力汇医慧影提升放射组学特征选择效率

■ 背景

运用放射组学，能进一步挖掘医学影像数据中蕴藏的信息，助力医疗机构更早、更快地发现细微病灶，从而将恶性疾病消弭于早期，来大幅减轻病患的痛苦，且有效提升医疗资源的使用效率，提升全民健康水平。而作为中国放射组学技术与解决方案的积极探索者，汇医慧影正以 AI 一体机等产品与平台，为医疗机构提供“全周期”、“一键式”的影像大数据科学分析能力，为放射组学技术在医疗机构的应用提供工具。

从前文所述可知，放射组学的基本流程分为数据收集、VOI 分割、特征提取、特征选择、模型训练以及模型评价及预测等步骤。由于放射组学的思路在于尽可能地提取医学影像中的更多数据特征，需要面对可能的“维度灾难”等问题。机器学习方法中的维度灾难，是指在样本量一定的情况下，随着输入维度的增加，空间数据会变得更为稀疏，这会严重影响模型的预测效果。要解决这一问题，则需要在特征选择阶段选择合适的算法对数据特征进行降维处理。

医疗机构部署放射组学方案需要通过大数据集进行训练，从而更精准地对患者的影像数据做出预测，此时就需要特征选择步骤具备更高的处理效率。因此，为方案配备更高处理能力的硬件基础设施，并需要根据算法特点进行针对性的调优至关重要。

因应这一需求，汇医慧影不仅引入了第二代英特尔® 至强 ® 可扩展处理器作为方案的强大处理引擎，而且采用面向英特尔® 架构优化的 Python 版本，来提升 LASSOCV、PCA 等特征选择算法的运行效率。

■ 方案与成效

LASSOCV、PCA 等算法是在基于放射组学技术的医学影像处理系统中，面向特征选择步骤的最常用算法，能够有效帮助系统缓解放射组学流程中常见的维度灾难问题，并使系统在压缩数据的同时让信息损失最小化，同时还有助于数据可视化，使信息呈现更直观。

汇医慧影在 AI 一体机配置了第二代英特尔® 至强® 可扩展处理器。该处理器不仅集成了更多的处理器内核和线程以及全面升级优化的微架构，也配备了更多高速缓存来提升处理效能，并可支持高达 36TB 的系统级内存容量；其内置的英特尔 ®AVX-512 带来的强大矢量计算能力，还能为放射组学方案中的模型训练和预测任务提供强劲的算力，让 LASSOCV 和 PCA 算法的执行更具效率。

同时，汇医慧影还与英特尔一起，针对算法执行语言 Python 进行了优化。由英特尔提供的面向英特尔® 架构优化的 Python，加入了对更多英特尔® 性能库（如英特尔® MKL）的支持，并内置了最新的矢量化指令。更为重要的是，其对 Scikit-learn（sklearn）库也有着良好的支持。

Sklearn 库是机器学习方法最常用的第三方库之一，对 LASSOCV 和 PCA 等常用机器学习算法进行了封装，同时也提供了 K-Fold 交叉验证等方法供用户方便调用。在面向英特尔® 架构优化的 Python 中的环境配置命令如下：

```
1. os.environ["KMP_BLOCKTIME"] = "0"
2. os.environ["USE_DAAL4PY_SKLEARN"] = "YES"
```

其中 KMP_BLOCKTIME 是设置某个线程在执行完当前任务并进入休眠之前需要等待的时间，此处设为 0 毫秒，USE_DAAL4PY_SKLEARN 是设置使用 SKLEARN 库。

与原生 Python 相比，面向英特尔® 架构优化的 Python 在特征选择的实际执行中有着巨大的效率提升。如图 2-5-7 上侧图所示，在勾选全部放射组学特征，采用 K-Fold 10 交叉验证的 LASSOCV 算法工作负载中，面向英特尔® 架构优化的 Python 执行速度是原生 Python 的 2.12 倍。而在下侧图中，勾选全部放射组学特征，采用 K-Fold 10 交叉验证的 LASSOCV+PCA 算法工作负载中，面向英特尔® 架构优化的 Python 执行速度是原生 Python 的 2.08 倍。

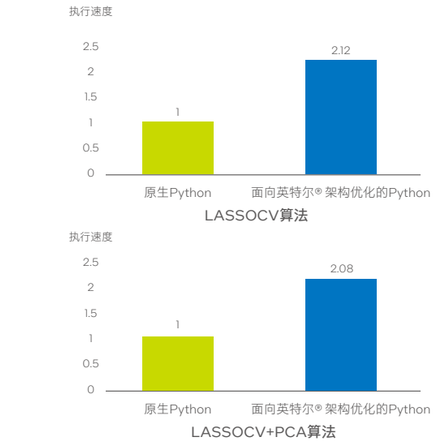


图 2-5-7 面向英特尔® 架构优化的 Python 与原生 Python 性能对比[48]

[48] 测试配置如下：处理器：双路英特尔® 至强® 金牌 6252 处理器，主频 2.1GHz，24 核心 48 线程；内存：192GB DRAM 内存；存储：INTEL SSDSC2BB48；BIOS 版本：SE5C620.86B.02.01.0009.092820190230；操作系统版本：18.04.1 LTS（Kernel：4.15.0-91-generi）；原生 Python 版本：Python2.7.17；面向英特尔® 架构优化的 Python 版本：Intel-Python2019U5；工作负载：由汇医慧影提供的医学影像分级训练

卫宁健康 NLP 后结构化平台提供由 AI 驱动的医疗信息整合解决方案

■ 案例背景

作为中国领先的医疗软件和解决方案提供商，卫宁健康一直以“科技赋能，提升人们健康水平”为使命，致力于成为“数字健康领域值得信赖的服务提供者”，构建的 NLP 后结构化平台，能够帮助医疗机构在一个联网数字平台上整合业务功能、医疗数据和服务交付流程。借助基于深度学习方法和机器学习方法的 AI 技术发展，卫宁健康正专注于在各个医疗领域中开发由 AI 驱动的解决方案，为临床医生和医院工作人员提供帮助。

卫宁健康 NLP 后结构化平台的设计目的是帮助医院整合多个医疗数据来源，包括临床记录、影像报告、实验室测试、探视记录等。例如，在健康信息系统 (HIS) 中，就包含了来自多个科室，由不同医生、护士、其他临床医生和助理输入的关于患者的多种非结构化数据，成了一个针对每个患者的非结构化、零散且不相关的海量数据库。卫宁健康平台希望通过 NLP 技术的引入与部署，为医疗机构提供一个面向患者数据的整体视图，从而帮助医生做出更精准的临床决策，并提供更好的患者治疗和研究。

以该平台在中国一家知名医院的应用为例，该医院正借助卫宁健康 NLP 后结构化平台开展一种在初期很难诊断出来的恶性疾病的研究和治疗。多学科会诊需要影像科、外科及其他学科的参与。而原始影像信息仅提供定性分析，而且过去没有与电子病历系统整合。

借助卫宁健康平台，相关的实体识别和关系提取模型可被用于评估病灶影像质量控制。通过这种技术，可以进一步分析影像诊断和病理结果，从而提高该恶性疾病诊断的准确性，并最终改善患者的预后。通过引入 NLP 技术并将相关能力集成到信息系统中，相关的关键信息可从报告中提取，并通过对多样化数据进行分析，实现在新的流程下的大规模数据自动分析，进而帮助医生做出准确的诊断并开展临床研究。

这样的方案无疑对平台算力以及 AI 加速能力提出挑战，为此，卫宁健康与英特尔展开合作，引入第四代英特尔® 至强® 可扩展处理器，借助英特尔® 高级矩阵扩展 (Intel® Advanced Matrix Extensions，英特尔® AMX) 和 16 位量子化技术，来为方案中的命名实体识别算法等提供优化，优化方案经实际验证后，被证明具有良好的效果。

■ 优化方案与成效

第四代英特尔® 至强® 可扩展处理器集成了用于助推 AI 能力的英特尔® AMX 和其他 AI 加速器。此外，针对英特尔® 架构和英特尔 AI 加速器而优化的软件也可以显著提升诸如卫宁健康 NLP 后结构化平台这样解决方案的性能。

卫宁健康NLP 解决方案在英特尔® 架构平台进行基准测试时，命名实体识别任务是基于 BERT 语言模型。最初的卫宁健康 NLP 模型采用 PyTorch 创建。以下优化措施用于在第四代英特尔® 至强® 可扩展处理器上提高推理吞吐量：

- 使用面向 PyTorch 的英特尔® 扩展优化框架 (Intel® Extensions for PyTorch，IPEX，可由英特尔® oneAPI AI 工具套件提供)，并进一步将 IPEX 用在卫宁健康代码中；
- 在第四代英特尔® 至强® 可扩展处理器上使用英特尔® AMX，进行自然语言处理加速；
- 基于 BF16 进行模型量化，在保证准确度的同时，与英特尔® AMX 结合，以实现矩阵运算性能的大幅提升。

为验证上述优化项的效果，卫宁健康与英特尔一起开展了相应的对比测试。测试在第三代英特尔® 至强® 可扩展处理器 (英特尔® 至强® 铂金 8380 处理器) 与第四代英特尔® 至强® 可扩展处理器 (英特尔® 至强® 铂金 8480+ 处理器) 之间展开，并评估了不同优化项对不同处理器平台性能的影响[49]。测试中，命名实体识别任务是基于 BERT 语言模型展开。

首先在英特尔® 至强® 铂金 8480+ 处理器的不同优化项对比上，如图 2-5-8 所示，在精度为 FP32 的数据类型下，优化后的工作负载在英特尔® 至强® 铂金 8480+ 处理器上的吞吐量，比未优化的工作负载 (基准值) 增加了 1.64 倍。另一方面，通过 IPEX 增强带来的优化，以及使用英特尔® AMX 加速矩阵计算和 BF16 量化共同发挥作用，使吞吐量综合增加至基准值的 6.04 倍。

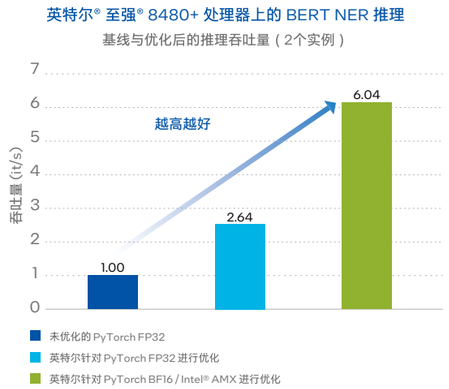


图 2-5-8 英特尔® 至强® 铂金 8480+ 处理器上的 BERT NER 推理结果比对

而在基于英特尔® 至强® 可扩展处理器代际运行的性能对比上，结果如图 2-5-9 所示，最新第四代英特尔® 至强® 可扩展处理器更有优势。使用英特尔® 至强® 铂金 8480+ 处理器且加入英特尔® AMX 和 BF16 量化优化，与完全未做优化的前一代英特尔® 至强® 铂金 8380 处理器相比，吞吐量增加了 6.3 倍。

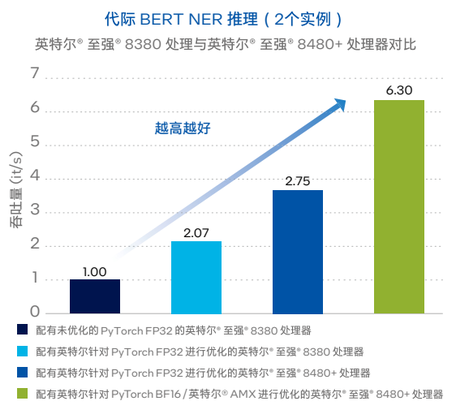


图 2-5-9 英特尔® 至强® 铂金 8380 处理器与英特尔® 至强® 铂金 8480+ 处理器对比

自 2021 年底部署以来，卫宁健康 NLP 后结构化平台解决方案已帮助相关医疗机构整合、链接并分析了 57,000 份相关疾病患者的病历。同时，平台也已经帮助研究人员根据研究对象提取并识别了 32 个影像特征和 114 个病理特征。

东软医保借力第四代英特尔® 至强® 可扩展处理器加速 OCR 票据识别

■ 案例背景

医疗保障 (医保) 在医疗系统整体运行中扮演着重要的角色。在传统的医保单据识别流程中，在无法联网结算时，医院需要将所有相关的住院、用药、就诊信息打印为纸质单据，并将纸质单据提交给医保结算柜台，医保机构随后会录入这些纸质单据中的信息并进行处理。传统模式的这一手动录入，不仅耗时耗力，而且还可能因为人为疏忽导致错录、漏录等问题。

为响应建设服务型政府号召，帮助医保部门提高医保结算效率，使医保经办人员摆脱重复性、事务性工作，实现精细化管理，东软推出了医保 OCR 票据识别方案。如图 2-5-10 所示，新方案能通过纸质单据电子化、OCR 文字识别、人工辅助校改、目录智能比对等流程，最终形成符合业务系统报销要求的医保电子结构化数据，从而降低人工成本、优化医保经办工作流程，保障医保基金安全。

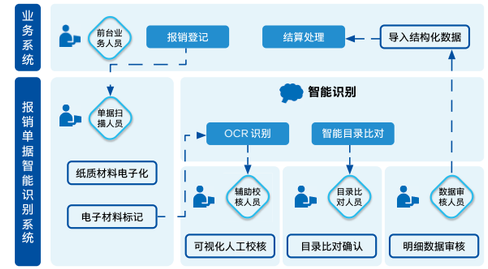


图 2-5-10 东软医保 OCR 票据识别方案应用流程

为解决方案中智能 OCR 票据识别在算力资源、成本等方面的挑战，东软采用了基于第四代英特尔® 至强® 可扩展处理器的服务器作为基础算力设备，并通过 OpenVINO™ 工具套件进行优化，实现了高性能、高性价比的 AI 推理。

■ 优化方案与成效

智能 OCR 是该方案的关键技术，为识别不同医院打印出的处方、明细、项目名称、数量和单价等信息，东软自研智能 OCR 算法，能够准确地在复杂背景下，识别出不同医院出具的不同格式单据，实现了较高的识别准确率。该方案在通过 OCR 将纸质单据转换为电子数据后，还会对数据进行智能化的匹配，以便于后续的数据处理。

[49] 基准配置 / 英特尔® 至强® 铂金 8480+ 处理器 (FP32) 上未优化的 PyTorch：测试由英特尔在 2022 年 10 月 17 日进行。单节点，双路英特尔® 至强® 铂金 8480+ 处理器 (2.0GHz)，112 核，开启超线程，开启睿频加速技术，512GB 总内存 (16 插槽 /32GB/4800MHz [运行频率 4800MHz])，BIOS：00.01.21，Ucode：0x2b000041，Ubuntu 22.04.1 LTS，5.15.0-48-generic，gcc 11.2.0，BERT 用于处理 NER 任务推理工作负载，框架：Pytorch 1.12.1，拓扑：Bert-Base-Chinese，数据集：JSON 格式 612 中文医疗报告，数据类型：FP32
基于英特尔® 至强® 铂金 8480+ 处理器 (FP32) 的 Intel® Optimization for PyTorch*：测试由英特尔在 2022 年 10 月 17 日进行。单节点，双路英特尔® 至强® 铂金 8480+ 处理器 (2.0GHz)，112 核，开启超线程，开启睿频加速技术，512GB 总内存 (16 插槽 /32GB/4800MHz [运行频率 4800MHz])，BIOS：00.01.21，Ucode：0x2b000041，Ubuntu 22.04.1 LTS，5.15.0-48-generic，gcc 11.2.0，BERT 用于处理 NER 任务推理工作负载，框架：Pytorch 1.12.1 + Intel® Extension for Pytorch 1.12.3，Intel OpenMP，Tcmalloc 2.10，OMP_NUM_THREADS=56，KMP AFFINITY=granularity=fine,compact,1,0，KMP_BLOCKTIME=1，拓扑：Bert-Base-Chinese，数据集：JSON 格式 612 中文医疗报告，数据类型：FP32
基于英特尔® 至强® 铂金 8480+ 处理器 (BF16) 的 Intel® Optimization for PyTorch*：测试由英特尔在 2022 年 10 月 17 日进行。单节点，双路英特尔® 至强® 铂金 8480+ 处理器 (2.0GHz)，112 核，开启超线程，开启睿频加速技术，512GB 总内存 (16 插槽 /32GB/4800MHz [运行频率 4800MHz])，BIOS：00.01.21，Ucode：0x2b000041，Ubuntu 22.04.1 LTS，5.15.0-48-generic，gcc 11.2.0，BERT 用于处理 NER 任务推理工作负载，框架：Pytorch 1.12.1 + Intel Extension for Pytorch 1.12.3，Intel OpenMP，Tcmalloc 2.10，OMP_NUM_THREADS=56，KMP AFFINITY=granularity=fine,compact,1,0，KMP_BLOCKTIME=1，拓扑：Bert-Base-Chinese，数据集：JSON 格式 612 中文医疗报告，数据类型：BF16
基准配置 / 英特尔® 至强® 铂金 8380 处理器 (FP32) 上未优化的 PyTorch：测试由英特尔在 2022 年 11 月 8 日进行。单节点，双路英特尔® 至强® 铂金 8380 处理器 (2.30GHz)，80 核，开启超线程，开启睿频加速技术，512GB 总内存 (16 插槽 /32GB/3200MHz [运行频率 3200MHz])，BIOS：SE5C6200.86B.0022.D64.2105220049，Ucode：0xd000375，Ubuntu 20.04.5 LTS，5.4.0-131-generic，gcc 9.4.0，BERT 用于处理 NER 任务推理工作负载，框架：Pytorch 1.12.1，拓扑：Bert-Base-Chinese，数据集：JSON 格式 612 中文医疗报告，数据类型：FP32
基于英特尔® 至强® 铂金 8380 处理器 (FP32) 的 Intel® Optimization for PyTorch*：测试由英特尔在 2022 年 11 月 8 日进行。单节点，双路英特尔® 至强® 铂金 8380 处理器 (2.30GHz)，80 核，开启超线程，开启睿频加速技术，512GB 总内存 (16 插槽 /32GB/3200MHz [运行频率 3200MHz])，BIOS：SE5C6200.86B.0022.D64.2105220049，Ucode：0xd000375，Ubuntu 20.04.5 LTS，5.4.0-131-generic，gcc 9.4.0，BERT 用于处理 NER 任务推理工作负载，框架：Pytorch 1.12.1 + Intel Extension for Pytorch 1.12.3，Intel OpenMP，Tcmalloc 2.10，OMP_NUM_THREADS=40，KMP AFFINITY=granularity=fine,compact,1,0，KMP_BLOCKTIME=1，拓扑：Bert-Base-Chinese，数据集：JSON 格式 612 中文医疗报告，数据类型：FP32

为实现高性能、低成本的 OCR 推理能力，东软选择第四代英特尔® 至强® 可扩展处理器作为方案的核心算力引擎。第四代英特尔® 至强® 可扩展处理器通过创新架构增加了每个时钟周期的指令，每个插槽多达 60 个核心，支持 8 通道 DDR5 内存，有效提升了内存带宽与速度，并通过 PCIe 5.0（80 个通道）实现了更高的 PCIe 带宽提升。

而在智能 OCR 所需的 AI 加速能力上，第四代英特尔® 至强® 可扩展处理器内置了创新的英特尔® AMX 加速引擎，其通过提供矩阵类型的运算，显著增加了人工智能应用程序的每时钟指令数（IPC），可为 OCR 工作负载提供显著的性能提升。同时，第四代英特尔® 至强® 可扩展处理器与 OpenVINO™ 工具套件的结合，可以进一步提升智能 OCR 所需的推理性能，因此智能 OCR 应用顺理成章，OpenVINO™ 工具套件成为东软智能 OCR 应用的 AI 框架。

方案在部署后，东软医保验证了 OCR 算法在第三代 / 第四代英特尔® 至强® 可扩展处理器上的代际性能对比，以及在不同精度的数据类型（FP32 / INT8）下的性能对比。

基于第三代 / 第四代英特尔® 至强® 可扩展处理器的 OCR 模型推理性能测试数据，如图 2-5-11 所示，在数据类型的精度同为 FP32 时，相比未采用矢量神经网络指令（VNNI）的第三代英特尔® 至强® 可扩展处理器，第四代英特尔® 至强® 可扩展处理器实现了 1.42 倍的性能提升 [50]。

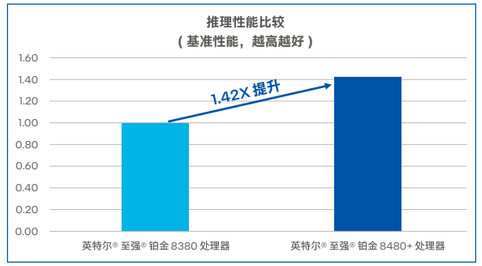


图 2-5-11 OCR 模型在第三代 / 第四代英特尔® 至强® 可扩展处理器上的推理性能对比

同时，东软利用第四代英特尔® 至强® 可扩展处理器的英特尔® AMX 加速器，将模型转换成 INT8 数据类型。如图 2-5-12 所示，转化后的模型推理性能结果与采用 VNNI 的第三代英特尔® 至强® 可扩展处理器相比，实现了 2.29 倍的性能提升 [51]。

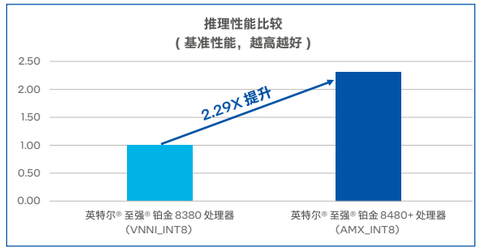


图 2-5-12 第三代英特尔® 至强® 可扩展处理器 + 英特尔® AVX – 512 _ VNNI 与第四代英特尔® 至强® 可扩展处理器 + 英特尔® AMX_INT8 性能对比

而在第四代英特尔® 至强® 可扩展平台中，不同精度（INT8 / FP32）的数据类型对比上，INT8 相比 FP32 实现了 4.66 倍的性能提升。

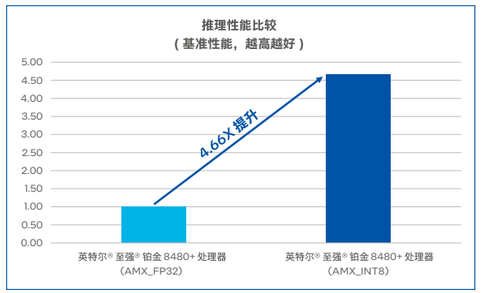


图 2-5-13 不同数据精度在第四代英特尔® 至强® 可扩展处理器上的推理性能比较

另外，经测试，也验证了东软医保 OCR 票据识别方案能够有效解决单据识别问题，且将处理时间缩短为传统手动流程的三分之一 [52]，为客户带来如下收益：

- **管理规范化**：实现单据处理的事务性工作和专业性工作分离，明确责任，落实公平、公正原则；
- **业务智能化**：AI+ 传统业务结合，OCR 识别准确度可达 95% 以上 [53]，缩短业务办理周期；
- **档案电子化**：档案业务一体化，减少纸质材料管理成本，提高复查、检索能力；
- **数据精细化**：搭建医疗知识库，使得目录对照越用越准，进而提高审计精细化程度，降低医保基金潜在风险。

目前，东软医保 OCR 票据识别方案已经在多家医保部门得到成功落地。以某市医保局为例，自方案正式上线运行以来，日均处理档案袋 20 个，累计处理单据 492 张，积累单据明细比对数据超过 30W，医保定制化目录对照经验库数据累计过百万，显著提高了医保业务的智能化水平 [54]。

小结

利用不同的 AI 方法，构建更为高效的慢性病预防和管理以及放射组学模型，通过更有效的疾病防治和病理检测方法，减少病患痛苦，提升全民健康水平。

在慢性病预防和管理上，第四范式与英特尔针对慢性病特征，推出了闭环的慢性病预防管理系统。一系列英特尔® 架构软硬件产品为之提供了强有力的计算与存储能力，使系统在慢性病预测等多种应用实践中都有着良好的表现。

在基于放射组学技术的医学影像处理方案中，汇医慧影与英特尔一起，携手打造基于机器学习方法的 AI 一体机，通过面向英特尔® 架构优化的 Python 对全新医学影像检测能力进行优化，帮助医疗机构有能力对早期恶性疾病病灶等实施检测。

而在卫宁健康 NLP 后结构化平台解决方案中，通过将解决方案集成到医疗机构的信息化系统中，分散的患者数据可以被智能地合并成为一个更全面的信息库。在采用英特尔® AMX 和 BF16 对英特尔® 至强® 铂金 8480+ 处理器进行优化后，该解决方案的性能得以改善，与基于英特尔® 至强® 铂金 8380 处理器的平台相比，命名实体识别推理的吞吐量提升达 6.3 倍[同][49]。加速推理可以帮助临床医生和研究人员更快地从多个临床部门的多种数据中获得洞察，从而实现更好的治疗效果。

最后，为帮助医保机构提升纸质单据的处理效率，释放人力资源，同时降低人工录入存在的信息疏漏等风险，东软推出了医保 OCR 票据识别解决方案。该方案能够通过由 AI 赋能的 OCR 应用，将相当一部分的医保票据识别转为自动化流程，可将处理时间缩短三分之二 [55]。为解决智能 OCR 票据识别在算力资源、总体拥有成本（TCO）等方面的挑战，东软采用了基于第四代英特尔® 至强® 可扩展处理器的服务器作为基础算力设备，并通过 OpenVINO™ 工具套件进行优化，实现了高性能、高性价比的 AI 推理。

[50] 截止 2022 年 8 月东软联合英特尔开展的测试。测试配置：基准配置 / 新配置 3 — 单节点，双路英特尔® 至强® 铂金 8380 处理器，40 核，开启超线程，开启睿频加速技术，256 GB 总内存（16 插槽 /16 GB/3200 MHz），<SE5C620.86B.01.01.0005.2202160810>，<0xd000375>，<Ubuntu 22.04.1LTS>，<5.19.0-051900-generic>，<gcc 11.2>，<Neusoft OCR>，<OpenVINO 2022.2.0-custom_onednn2.6_9a3a3181e7056dcf7ccd3a16e599e6882a4edc23>，<Neusoft OCR>，<OneDNN 2.6>；新配置 1/2 — 单节点，双路英特尔 ® 至强 ® 铂金 8480+ 处理器，56 核，开启超线程，开启睿频加速技术，256 GB 总内存（16 插槽 /16 GB/4800 MHz），<EGSDCRB1.SYS.0085.D15.2207241333>，<0x2b000070>，<Ubuntu 22.04.1LTS>，<5.19.0-051900-generic>，<gcc 11.2>，<Neusoft OCR>，<OpenVINO 2022.2.0-custom_onednn2.6_9a3a3181e7056dcf7ccd3a16e599e6882a4edc23>，<Neusoft OCR>，<OneDNN 2.6>。

[51] 截止 2022 年 8 月东软联合英特尔开展的测试。测试配置：基准配置 / 新配置 3 — 单节点，双路英特尔® 至强® 铂金 8380 处理器，40 核，开启超线程，开启睿频加速技术，256 GB 总内存（16 插槽 /16 GB/3200 MHz），<SE5C620.86B.01.01.0005.2202160810>，<0xd000375>，<Ubuntu 22.04.1LTS>，<5.19.0-051900-generic>，<gcc 11.2>，<Neusoft OCR>，<OpenVINO 2022.2.0-custom_onednn2.6_9a3a3181e7056dcf7ccd3a16e599e6882a4edc23>，<Neusoft OCR>，<OneDNN 2.6>；新配置 1/2 — 单节点，双路英特尔® 至强® 铂金 8480+ 处理器，56 核，开启超线程，开启睿频加速技术，256 GB 总内存（16 插槽 /16 GB/4800 MHz），<EGSDCRB1.SYS.0085.D15.2207241333>，<0x2b000070>，<Ubuntu 22.04.1LTS>，<5.19.0-051900-generic>，<gcc 11.2>，<Neusoft OCR>，<OpenVINO 2022.2.0-custom_onednn2.6_9a3a3181e7056dcf7ccd3a16e599e6882a4edc23>，<Neusoft OCR>，<OneDNN 2.6>。

[52] 数据援引自东软内部测试结果，通过对比传统手工报销流程（30 分钟）和新模式下报销流程（10 分钟）计算得出。英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。

[53] 数据援引自东软提供的信息。英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。

[54] 数据援引自东软提供的信息。英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。

[55] 数据援引自东软内部测试结果，通过对比传统手工报销流程（30 分钟）和新模式下报销流程（10 分钟）计算得出。英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。

基于联邦学习的 AI 方法在医疗行业中的探索

打破数据壁垒，提升医疗 AI 应用效能

利用多源数据提升训练性能

从前述内容可以看到，利用深度学习、机器学习等方法，AI 能有效提升医疗行业中医学影像处理、辅助诊断、疾病预测以及药物研发等领域工作的效率，帮助医生更全面、精确地了解病情，让病患早日摆脱病魔。除了选择合适的算法和需要充沛的算力，AI 效率的提升还有赖于更多的数据，来进行训练和推理验证，以提升模型准确率。尤其在图像分割、病理切片分析等应用方向，所使用的深度学习模型更需要大量样本数据来进行训练，才能达到较好的泛化能力（Generalization），并防止过拟合（Overfitting）。

为获取隐藏在数据像素后面的大量特征，医疗影像常用的深度学习模型一般会采用多层网络的方法，典型如卷积神经网络，在输入层和输出层之间有很多隐藏层，隐藏层的数量决定了学习的深度。模型中的一些学习方式，例如反向传播，会将输出与训练数据的误差进行比较，进而计算输出中的误差，而后相关的隐藏层会调整其权重来降低错误率。

因此，深度学习通常需要大量不同实例的数据集，让模型能从中学习到所需的特征，并生成带有概率向量的输出。所处理的图像越复杂，训练所需的数据量也越大。研究表明，如图 2-6-1 所示，传统机器学习方法中，AI 性能初期会随着训练数据量的增加而增长，后期则趋于平缓；而深度学习方法的性能则一直会随着训练数据量的增加而增长[56]。因此，为医疗行业 AI 应用，尤其是基于深度学习的 AI 应用提供更多不同实例的数据集，可以有效提升其性能。

同时在医疗科研领域，对数据资产的利用程度也会影响到科研效率。数据集的体量越大、维度越丰富，能够从中发现和学习到的特征就越多，由此构建的 AI 模型的性能及应用价值也就越高。大量统计数据已表明，有着多数据源融合与协作的医疗机构的科研效率往往会高于单一数据源的机构。因此，医疗科研机构普遍期望能开展多方及多样化的数据协作，来获取以下关键优势：

- **消除或降低数据偏差**：研究区域以及方法、方式的不同，会带来不同医疗机构间的数据差异，通过数据融合能消除或降低数据偏差，使研究成果泛化能力更强；
- **扩大科研样本量**：数据融合能够让不同研究中心的临床数据得以共享，进而扩大科研所需的数据样本量，提升最终 AI 模型的性能；
- **补充非临床数据**：许多长期跟踪的医疗科研数据还需要与社区医疗、家庭医生、体检机构，以及可穿戴设备的数据实施融合。

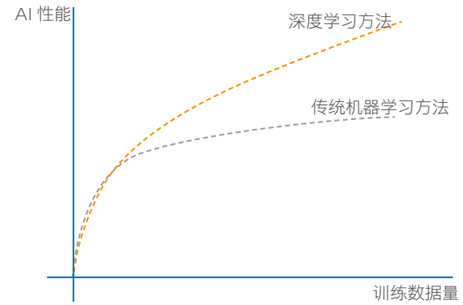


图 2-6-1 训练数据量对不同学习方法的影响

但与大多数行业一样，数据在医疗行业中“数据孤岛”问题同样严重，不同医疗机构，甚至不同科室的数据往往并不相互联通。而要做到完全的互联互通式数据共享，又势必面临如何保护数据隐私和安全的问题。众所周知，健康状况等数据是极为重要的个人隐私信息，如果因使用不当带来泄露风险，无疑是医疗机构无法接受的。而在国家政策层面，《个人信息保护法》、《数据安全法》等一系列法律法规的出台，也对数据安全和隐私信息保护，做出了明确和严格的规范。

为了向 AI 应用提供更多源、合规以及更高质量的训练数据集，许多科研与学术机构也提出了多种联合学习方法，例如机构增量学习（Institutional Incremental Learning，IIL）、循环机构增量学习（Cyclic Institutional Incremental Learning，CIIL）以及近年来声名鹊起的联邦学习（Federated Learning，FL）。

IIL 方法是让参与训练的各方顺序排列，训练模型按顺序依次传递，前一参与方用自己的数据训练模型进行训练，然后将结果传递给后一参与方，后一方再用自己的数据重新训练，而 CIIL 是在 IIL 的基础上多次循环迭代。这两种方法在实践中都存在一些缺陷，首先是它们都采用了共享模型的方式，训练模型需要在不同参与方之间传递，容易造成隐私泄露和数据安全问题；其次，这两种方法每个参与方的训练数据如果过小，例如每个医疗机构只能提供数名患者的数据，那联合学习的效果并不能得到有效改善；最后，以上的方法都采用一种串行协作模式，需要将模型完全传递给下一个参与方，对网络性能也有一定的要求。

[56] 该观点由 Zhu, X. et al., Do we Need More Training Data?https://arxiv.org/abs/1503.01508, March 2015. 、Shchutskaya, V., Latest Trends on Computer Vision Market, https://indatalabs.com/blog/data-science/trends-computer-vision-software-market?cli_action=1555888112.716 以及 Why go large with Data for Deep Learning? https://towardsdatascience.com/why-go-large-with-data-for-deep-learning-12eee16f708?gi=ba92e606d0 等文综合得出。

与以上两种方法不同，联邦学习方法则使用并行的协作方法，使得参与联合学习的各方都在本地使用本地化数据对模型进行训练，然后再将训练得到的模型参数进行共享。这能带来显而易见的优势，一方面，各方的训练数据和模型都留在了本地，在数据安全和隐私保护方面有了更好的保障；另一方面，并行的训练模式使训练效果获得叠加，有效提升了训练效果。同时由于在并行协作方法中，数据和模型与训练的结合接近分布式训练，因此训练效率要高于串行协作方法。

构建联邦学习系统的核心，是为各参与方打造可信数据共享方式。目前，基于硬件可信执行环境（Trusted Execution Environment，TEE）技术的解决方案正越来越受到医疗行业的青睐。其核心理念是以第三方硬件为载体，为不同数据源提供安全可信高效的计算环境。如图 2-6-2 所示，来自 A、B 不同数据源的训练优化结果，可以在右侧由硬件创建的 TEE 环境中进行共享，并生成最终的优化模型。在各种 TEE 方案中，英特尔® 软件防护扩展（Intel® Software Guard Extensions，英特尔® SGX）是目前较为成熟，且广受用户好评的方案。

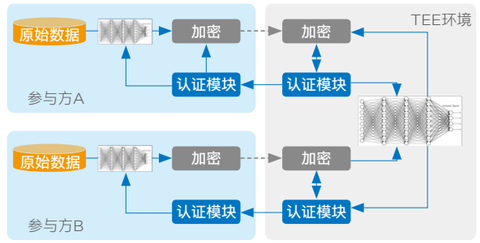


图 2-6-2 联邦学习中的 TEE 环境

基于联邦学习的 AI 方法

联邦学习根据使用场景的不同，可分为横向联邦学习（Horizontal Federated Learning）、纵向联邦学习（Vertical Federated Learning）以及联邦迁移学习（Federated Transfer Learning）等。其中横向联邦学习适用于数据集中，特征重叠较多，而用户重叠较少的情况。其可以将数据集按用户维度切分，并取出特征相同而用户不完全相同的数据进行训练。例如，在同一种病理图像处理中，来自不同医疗机构的用户数据，就可以按照横向联邦学习方式进行训练。

纵向联邦学习则适用于不同数据集中，用户重叠较多而特征重叠较少的情况。这一模式可以将数据集按照特征维度切分，并取出用户相同而特征不完全相同的那部分数据进行训练。典型场景例如对病患进行结构化病理诊断，同一批用户在不同检查项中的数据，就可以按照纵向联邦学习方式进行训练。而联邦迁移学习是在用户和特征重叠均较少的情况下，不对数据进行切分，而利用迁移学习的方法来完成数据联合训练。

以使用 AI 方法进行病理图像分割的场景为例，医疗机构 A、B 各自拥有大量的患者的病理图像资料，出于安全隐私考虑，这些图像数据存在于各自的数据中心，并通过防火墙实施了高等级隔离，任何直接的数据访问都会被拒绝。

在通过联邦学习的方式来训练这两组数据的过程中，为保证训练过程中的数据保密性，如图 2-6-3 所示，需要借助协同方 C 进行加密训练。加密训练过程分为以下步骤：

1. 协同方 C 把公钥分发给 A 和 B，用以对训练过程中需要交换的数据进行加密；
2. A 和 B 之间互相以加密的形式交互用于计算梯度的中间结果；

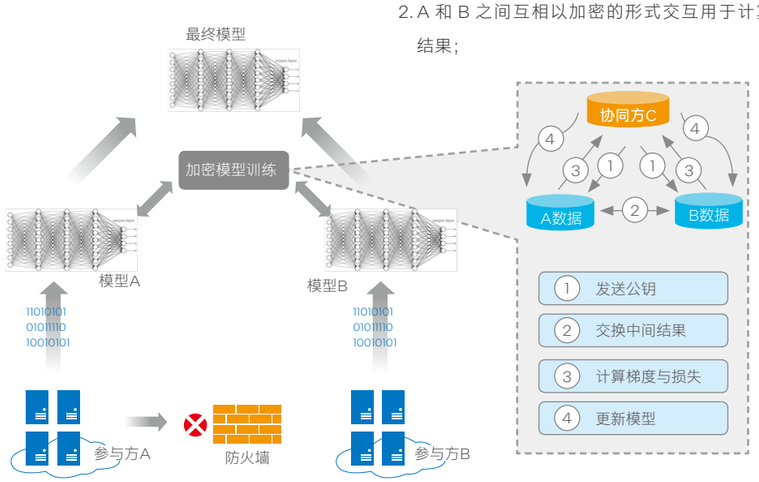


图 2-6-3 联邦学习基本架构

3. A 和 B 分别基于加密的梯度值进行计算，并将结果汇总给协同方 C。协同方 C 通过汇总结果计算总梯度值并进行解密；
4. 协同方 C 将解密后的梯度分别回传给 A 和 B，A 和 B 再以此更新各自模型的参数。

上述训练迭代步骤将一直持续至损失函数收敛，训练过程完成并得到最终的模型。联邦学习所传递的参数包括了：

- 深度学习架构的典型超参数，例如 Batch Size，优化器，学习率等；
- 每轮学习的 Epochs（EpR），更多 EpR 可以加速收敛，但收益递减；
- 每轮学习中的参与者数量；
- 模型更新所使用的压缩 / 修剪方法。

与一般的分布式机器学习 / 深度学习方法相比，联邦学习方法具有以下特征：

- 数据不脱离本地：参与者利用自身拥有的数据训练全局模型；
- 每个参与方都参与学习过程，模型损失可控；
- 训练过程中兼顾隐私和安全，参与各方能够在不披露底层数据及其加密形态的前提下共建模型。

除此之外，联邦学习还具有良好的效果激励机制，即通过联邦学习建立模型后，模型的效果能够获得评估，并通过永久数据记录机制进行记录。提供高质量数据多的参与方所获得的模型效果会更好，模型效果取决于数据提供方对自己和他人的贡献。这些模型的效果在联邦效果激励机制上会分发给各数据源，以此获得联邦的奖励，并继续激励更多数据源加入联邦。

基于以上特点，联邦学习能为医疗行业 AI 应用提供跨机构、跨部门的数据共享方法和模型训练方式，帮助实现各数据源的私密数据不出本地，只通过加密机制下的参数交换，在不违反数据隐私法规的情况下建立学习模型优化机制。

联邦学习源码可参考：https://www.tensorflow.org/federated/

英特尔® 软件防护扩展（英特尔® SGX）

技术简介

作为 TEE 方案技术实现的典型代表，英特尔® SGX 通过一组新的指令集扩展与访问控制机制，在硬件（例如内存）中构造出一个可信的“飞地”（Enclave），使数据和应用程序的安全边界仅限于飞地本身以及处理器内，实现不同应用程序间的隔离运行。同时其运行过程也可不依赖于其他软、硬件设备。这意味着数据的安全保护是独立于软件操作系统或硬件配置，即便在硬件驱动程序、虚拟机乃至操作系统均受到攻击破坏的情况下，也能杜绝数据泄露和篡改，从而增强应用程序代码和数据的安全性。

传统上，数据的隐私保护和安全防护大都是工作在操作系统或软件层面，但是当操作系统或软件受到“感染”时，数据的安全性就变得岌岌可危。如图 2-6-4 所示，虽然应用程序可以通过安全扫描，防火墙等对来自外部黑客或应用程序的攻击进行防护，但是恶意软件、恶意代码如果利用操作系统漏洞，就可以绕过这些防护，直接攻击关键的隐私数据。

因此，英特尔® SGX 可以为用户提供更强的安全防护，并具备以下主要特性：

- **增强的保密性和完整性**：飞地工作在隔离的硬件环境（支持 SGX 技术的英特尔® 架构处理器、内存）中，并通过密钥对应用系统和数据实施鉴权，即使在操作系统、BIOS 或虚拟机等中存在高权限恶意软件或恶意代码，也无法对数据实施攻击；
- **更小的安全攻击面**：英特尔® SGX 将应用程序与敏感数据限定运行在受保护的硬件飞地中，杜绝了传统上恶意程序可能从硬件、虚拟机和操作系统发起的攻击，更小的攻击面带来了更高的安全性；

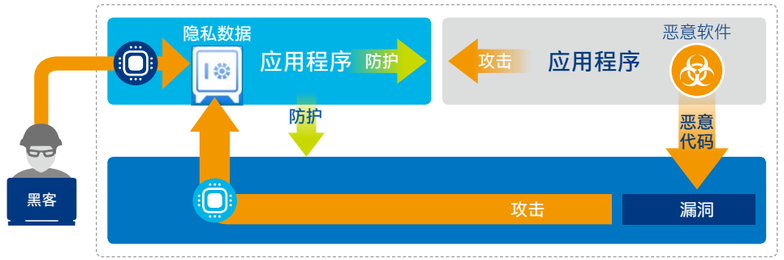


图 2-6-4 被实施内部攻击的应用程序

- **远程鉴权和控制能力**：用户可以通过执行远程鉴权，更安全地将密钥、凭据和其他敏感数据提供给飞地；
- **增强的联邦学习效率**：在基于英特尔® SGX 技术执行的联邦学习过程中，AI 模型和训练数据都部署在受保护的硬件飞地中，大幅降低因应用程序和数据加解密带来的通信和计算成本，使学习效率更高；
- **更低学习曲线**：采用英特尔® SGX 技术的应用程序可基于特定英特尔® 架构处理器平台进行开发、集成和执行，开发人员只需安装相关驱动并进行 SDK 适配，无需熟悉额外的软硬件环境，编程方式也无需更改，学习曲线更低。
- **更加高效的实现**：与基于安全多方计算，同态加密等技术的联邦学习实现方法相比，基于英特尔® SGX 技术的硬件 TEE 方案运行效率更高。

英特尔® SGX 安装与配置

用户可以通过引入英特尔® SGX SDK 来创建基于英特尔® SGX 的解决方案，该 SDK 提供了以下内容：

- API
- 函数库
- 文档
- 样本源码
- 工具

可以访问以下链接获得最新的英特尔® SGX SDK:
基于 Windows 系统的 SDK 下载地址
https://registrationcenter.intel.com/en/forms/?productid=2614
基于 Linux 系统的 SDK 下载地址
https://01.org/intel-software-guard-extensions/downloads

基于英特尔® SGX 的典型解决方案

借助英特尔® SGX，医疗机构可以根据自身需求来构建多样化的解决方案。下文将简单介绍一种典型的基于英特尔® SGX，协同方采用中心聚合服务器（Aggregator）的多源数据 AI 模型训练解决方案。

解决方案架构如图 2-6-5 所示，采用位于中心的聚合服务器 “飞地” 以及部署在不同参与方的边缘 “飞地” 组成网络。聚合服务器和各参与方中的 “飞地”，均是由英特尔® SGX 提供的处理器指令，可在内存中构造出具有高等级安全访问权限的可信区域。

方案中，在加密通道中被传输的是 AI 模型的各种参数，而训练数据、明文 AI 模型以及 AI 算法则被留存在各个节点本地。在初始化过程中，各 “飞地” 首先产生公私密钥对，公钥注册到聚合服务器，私钥保存在各自的 “飞地” 里。当训练开始时，聚合服务器会先和目标 “飞地” 建立基于对称加密密钥的连接。连接建立后，聚合服务器会先将待训练的模型共享参数加密推送到各个 “飞地” 中，然后各 “飞地” 把模型参数解密传送到本地 AI 训练环境对本地数据实施训练。训练结束后，本地 AI 训练环境将训练得到的共享参数返回至本地的 “飞地”。

以上 “飞地” 间的传递流程可以进行多轮循环迭代，直至获得满意的训练结果，同时方案也可对各参与方的训练效果贡献度进行评估。

由于上述过程都是在 “飞地” 中实现，即在方案的整个循环迭代过程中，AI 模型参数都在加密通道以及 “飞地” 内进行传递和交互，并不与外界软、硬件接触，故而形成了安全可信的 “内循环”。同时，AI 模型和训练数据都留存在各个受保护的硬件飞地中，需要在加密通道中传递的只有中间参数，这无疑大为增加了联邦学习的执行效率。而基于英特尔® 架构的处理器，特别是第二代英特尔® 至强® 可扩展处理器，可为 “飞地” 的构建、加密通道的铺设以及中间参数交互和聚合提供强大算力。

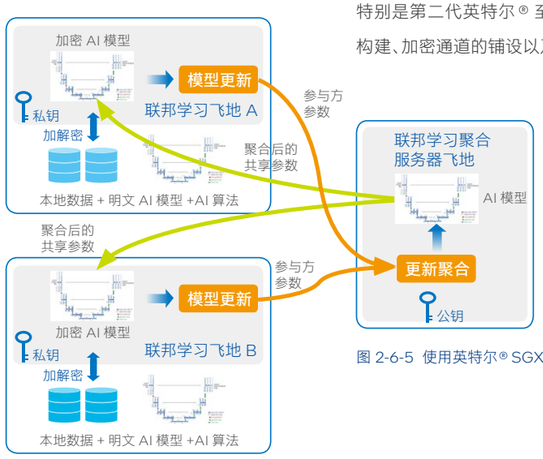


图 2-6-5 使用英特尔® SGX 的联邦学习方案

联邦学习在医疗领域的实战

基于联邦学习，开展面向脑部病灶分割的研究

■ 案例背景

深度学习方法一直是医疗图像处理领域的热门话题，在近年的国际医学图像计算和计算机辅助干预会议[57]（International Conference on Medical Image Computing and Computer Assisted Intervention, MICCAI）中，也不断有新的方法涌现。

让更多数据参与训练，例如创建公共可用的高质量多源数据集，用于基准测试和定量评估，能进一步提升影像处理性能已成为一种共识，但在实际运行过程中仍面临着巨大的挑战。一方面，如何将数据共享到集中位置仍需要解决系统架构和传输效率的问题；另一方面，与普通摄影图像相比，由于法律、隐私、技术和数据所有权等方面的限制，医疗数据的可用性更加有限。

从 2018 年开始，英特尔就开始与宾夕法尼亚大学生物医学图像计算与分析中心（CBICA）一起，就联邦学习在医疗影像处理上的应用展开联合探索，并形成了有效的应用实践，其成果可参阅相关论文《Ulti-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation》。下文将就该实践中，如何使用相关数据集，通过在聚合服务器上迭代聚合本地训练的模型，在不共享任何患者数据的情况下，应用联邦学习方法构建一个有效的图像分割模型，并使模型可为多个参与方提供服务的过程，进行简要描述。

■ 案例描述与成效

如图 2-6-6 所示，本案例采用了一个深度卷积神经网络（CNN）的 U-Net 拓扑，该模型将单道图像作为输入，并输出等效的二进制掩码，其会为每个像素分配一个类别标签。该网络模仿自动编码器的体系结构，其能够通过最大池化，具有捕获上下文的收缩路径，并通过上采样实现本地化的扩展路径。与标准的自动编码器不同，扩展路径中的每个特征图谱与来自收缩路径的对应特征图谱以跳跃连接（skip connection）的方式进行关联，这使得模型通过较小的感受域就能获取更多拥有空间信息的下游特征图谱。直观来说，这允许该网络考虑不同空间尺度下的特征。

现在，U-Net 已成为用于医学图像分割的标准深度学习拓扑之一，在神经超声图像分割、肺 CT 扫描影像分割等工作负载中发挥了巨大作用。本案例中联邦学习的各项验证测试均使用了该模型，其 Dropout 参数设为 0.2，上采样设置为真。更多 U-Net 分割网络的优化方法，请参阅前文相关介绍。

本方案中使用了 BraTS 2018 训练数据集[59]，其中包含了来自多个医疗机构确诊患者的多模式脑部扫描磁共振成像（MRI）。每个脑部扫描的放射线照相异常区域已用三种截然不同的标签进行手动标注。由于本案例是为了评估联邦学习在临床图像分割中的表现，因此只关注被以上三种标签标记为病灶的体积。同时，案例还选择了 IIL 和 CIIL 等联合学习方法作为对比组。

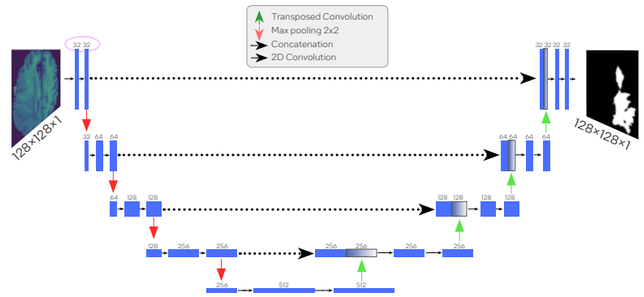


图 2-6-6 用于 BraTS 联邦学习方案的 U-Net 拓扑[58]

[57] 具体请参阅 http://www.miccai.org/
[58] 图片引用自 Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation，Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, Spyridon Bakas，https://arxiv.org/pdf/1810.04304v1.pdf
[59] 数据集引自 Menze, B.H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J.,Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M.A.,Arbel, T., Avants, B.B., Ayache, N., Buendia, P., Collins, D.L., Cordier, N., Corso,J.J., Criminisi, A., Das, T., Delingette, H., , D., Durst, C.R., Dojat, M., Doyle,S., Festa, J., Forbes, F., Geremia, E., Glocker, B., Golland, P., Guo, X., Hamamci,A., Iftekharuddin, K.M., Jena, R., John, N.M., Konukoglu, E., Lashkari, D., Mariz,J.A., Meier, R., Pereira, S., Precup, D., Price, S.J., Raviv, T.R., Reza, S.M.S.,Ryan, M., Sarikaya, D., Schwartz, L., Shin, H.C., Shotton, J., Silva, C.A., Sousa,N., Subbanna, N.K., Szekely, G., Taylor, T.J., Thomas, O.M., Tustison, N.J., Unal,G., Vasseur, F., Wintermark, M., Ye, D.H., Zhao, L., Zhao, B., Zikic, D., Prastawa,M., Reyes, M., Leemput, K.V.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Transactions on Medical Imaging 34(10), 1993-2024(2015)、Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J.S., Freymann,J. B., Farahani, K., Davatzikos, C.: Advancing The Cancer Genome Atlas gliomaMRI collections with expert segmentation labels and radiomic features. Nature Scientific Data 4, 170117 (2017) https://doi.org/10.1038/sdata.2017.117、Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J.,Davatzikos, C.: Segmentation Labels and Radiomic Features for the Pre-operativeScans of the TCGA-GBM collection. In: The Cancer Imaging Archive, (2017) 以及 Bakas, S., Akbari, H., Sotiras, A., Bilello, M., Rozycki, M., Kirby, J., Freymann, J.,Davatzikos, C.: Segmentation Labels and Radiomic Features for the Pre-operativeScans of the TCGA-LGG collection. In: The Cancer Imaging Archive, (2017)

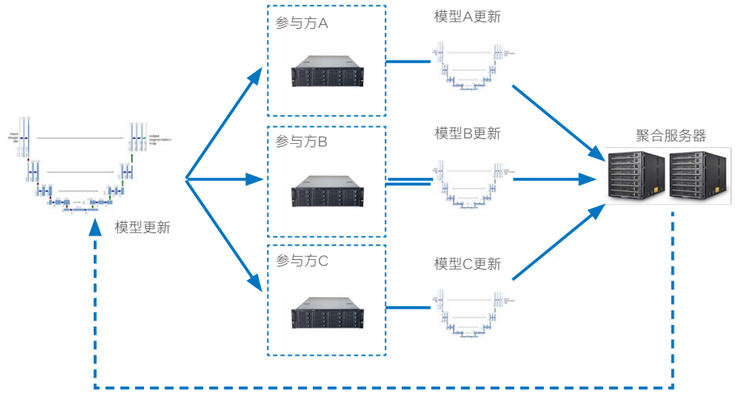


图 2-6-7 用于 BraTS 的联邦学习方案架构

联邦学习架构如图 2-6-7 所示，参与者都不需要共享各自的数据而是在本地训练共享模型，且仅将模型的更新发送到聚合服务器。聚合服务器整合更新的内容，并将新的共享参数发送给各参与方，以便进行进一步训练（可循环进行）或应用。整合的共享参数相当于各参与方更新的加权平均值，特定参与方的权重作为驻留在该参与方的总数据实例的分数给出。这一本地训练，更新整合和新参数分发的迭代过程被称为联合轮。方案中，对不同数量的参与者，以及不同的 EpR 对最终 AI 应用的性能影响进行了评估。更多联邦学习方案流程细节，请参阅第 72 页“基于英特尔® SGX 的典型解决方案”相关描述。

BraTS 图像分割效果可通过 Dice 系数（Dice Coefficient，DC）值来进行评估，其反映了预测与实际联合的交集比，可定义为：

$$DC = \frac{2|P \cap T|}{|P| + |T|}$$

其中 P，T 分别为预测和 GT（Ground Truth）的 Mask。方案中的基准值是通过 U-Net 拓扑，由经过完全共享（Data-Sharing）的数据训练得出。其经过验证的峰值精度 DC=0.862（最优值）。如图 2-6-8 所示，上侧表示各种联合学习方法在各联合轮中的 DC 值变化。可以发现，联邦学习方法的 DC 值最为稳定，且接近于完全数据共享方法下得到的最优值，而 IIL 和 CIIL 方法则波动幅度较大。下侧是表示在每次通过完整训练之后，各种联合学习方法的验证 DC 值，联邦学习方法的 DC 值也接近于最优值且非常稳定。

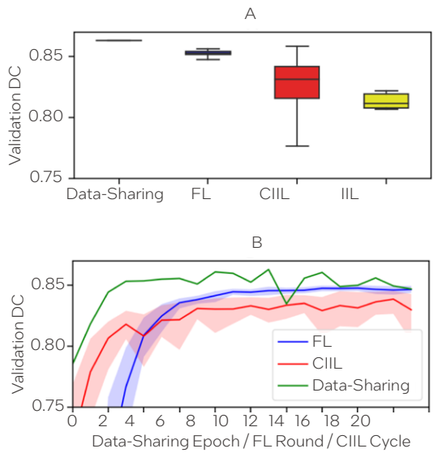


图 2-6-8 联邦学习与其他学习方式性能比较[60]

验证测试的结果表明，在医疗机构中使用联邦学习方法，其性能可以达到完全数据共享方法的 99%[61]，即使对于不平衡的数据集也是如此。无疑，通过引入联邦学习方法，医疗机构可以更有效地改善和提升计算机辅助分析和诊断系统的性能，从而促进精准医学的发展，同时也能有效应对一系列因数据共享产生的安全、隐私或数据所有权问题。

更多案例详情，请参阅：Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation，Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, Spyridon Bakas
https://arxiv.org/pdf/1810.04304v1.pdf

[60] 图片引用自 Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation，Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, Spyridon Bakas，https://arxiv.org/pdf/1810.04304v1.pdf

[61] 数据援引自 Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation，Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, Spyridon Bakas，https://arxiv.org/pdf/1810.04304v1.pdf

医渡云打造基于联邦学习的多方安全计算解决方案

■ 案例背景

为帮助众多医疗科研机构打造兼顾高效和安全需求的多方隐私计算能力，为医疗和健康行业提供更优的数据融合与数据科研价值挖掘能力，多年来一直深耕医疗 AI 与大数据技术创新的医渡云，以强大的医学数据治理能力为后盾，通过自研 YiduManda 安全计算引擎为数据融合提供了联邦学习、联合统计、联盟区块链等核心技术保障。

这其中，基于硬件可信执行环境（TEE）的联邦学习方法凭其在数据“可用不可见”方面的独到优势，在各医疗科研机构的实践中收获了良好效果，与其他多方隐私计算方案相比，展现出以下优势：

- 医疗数据不脱离本地，各参与方可利用自身拥有的数据训练全局模型；
- 每个医疗科研参与方都可参与训练过程，模型损失可控；
- 训练过程能更好地兼顾隐私和安全需求，各参与方能在不暴露数据及加密形态的前提下进行联合建模。

为此，医渡云与英特尔携手，引入英特尔® SGX，来为联邦学习方法应用打造基于硬件的可信执行环境（TEE）的联邦学习方法，来为各医疗科研机构打造提供高效的多方安全计算解决方案。

■ 案例描述与成效

医渡云基于联邦学习等隐私计算方法打造的多方安全计算解决方案，其功能层面如图 2-6-9 所示，自下而上分别是面向院内外业务系统的数据采集系统、进行数据加工治理的专病库以及开展多方隐私计算的安全计算平台。在安全计算平台之上，医渡云又通过多中心医学研究全场景解决方案，部署了一系列面向多样化医疗科研场景所需的上层应用能力，如临床研究开展、药械试验与研究、诊疗技术开放推广、患者随访与管理等。

图 2-6-9 医渡云多方安全计算解决方案整体架构

在处于核心位置的多方安全计算平台中，医渡云通过自研的YiduManda，以多方安全计算、联邦学习为基础，同时结合英特尔® SGX，将来自各个科研参与方（医院）的原始数据，通过联合统计、特征工程（Feature Engining）、逻辑回归（LR）、XGBoost 等方法进行联合统计分析和模型训练，并最终得到医疗科研 AI 模型以及相关深度学习模型。

在架构设计上，医渡云的方案采用了分布式的设计，如图 2-6-10 所示可分为平台端（调度节点）和医院端（计算节点），其中：

- **平台端（调度节点）**：部署在云环境或机构联盟的主中心私有云环境中，包括一套用于联邦学习等隐私计算的调度层框架以及相应的科研应用平台。应用层框架对各医院端隐私计算节点进行统一的管理和协调，并对多方安全计算任务进行统一调度；
- **医院端（计算节点）**：部署在医院的私有云环境中，通过隐私计算节点间的协作，能保证数据在不出医院的前提下完成联邦学习等多方隐私计算过程，且各个节点对其所有的数据有绝对控制权，所有数据调用经过多方安全计算框架可审计。

基于上述功能与架构设计，各医疗科研机构之间可基于联邦学习开展模型协同训练。在数据准备阶段，数据准备和预处理工作是在各个参与协同训练的医院或医疗机构本地完成的，准备好的数据可通过程序接口加载到医院端，随后平台端会调度完成模型的协同训练过程。参与训练的医院端通过加密信道与其它参与方完成通信和计算，并最后完成模型的优化训练。

在方案的具体部署中，引入了英特尔® SGX 为联邦学习构建基于硬件的可信执行环境（TEE）。目前，包括第三代英特尔® 至强® 可扩展处理器、第四代英特尔® 至强® 可扩展处理器等平台都已集成了英特尔® SGX，其能在内存的特定硬件环境中构造出一个可信的安全“飞地”（Enclave），为医疗科研过程中参与多方计算的敏感数据和代码提供更强的安全防护。

与其它技术方案相比，英特尔® SGX 一方面可为敏感数据与程序构建隔离的硬件环境，使安全保护机制独立于软件应用、操作系统或硬件配置之外，从而令保密性和完整性大幅提升；另一方面，独立的“飞地”设置可让关键的应用程序和数据更有效地避开来自硬件驱动程序、虚拟机乃至操作系统的攻击，带来更强的安全性。基于英特尔® SGX 提供的这些优势，各医疗科研机构就可将数据分析、模型训练及推理所涉及的数据运行在“飞地”中，通过访问控制为这些应用代码和数据提供更可信赖的安全保障。

而在性能表现上，英特尔® SGX 基于硬件层面的安全保护机制，可使敏感数据与应用程序获得来自基于英特尔® 架构处理器强劲性能的加速或助推，从而更好地解决方案中性能和安全的平衡问题，在某些对计算性能和安全等级要求都很高的医疗科研场景中，打造更为全面的应用优势。

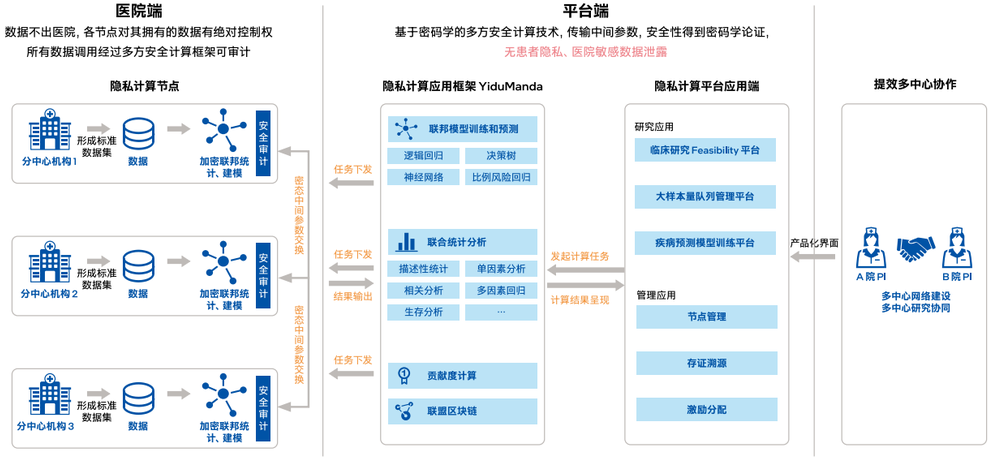


图 2-6-10 医渡云多方安全计算解决方案中医院端和平台端的协作模式

锘崴科技开展基于隐私保护计算的 GWAS 研究

■ 案例背景

全基因组关联分析（Genome-Wide Association Studies，GWAS）一直是生物医学领域开展各项研究的重要方法，其是指从人类全基因组范围内找出存在的序列变异（即单核苷酸多态性（Single Nucleotide Polymorphisms, SNPs）），并筛选出与疾病相关的 SNPs 来帮助开展诊断或预防。这一方法常用于一些复杂疾病的研究。这类疾病往往受多个基因和环境因素共同影响，每个基因的单独作用较弱，且往往存在多基因间和基因环境间的交互作用，因此被称作复杂疾病。利用 GWAS 对其遗传机制的研究有助于开发新药物、发展新疗法和开展预防工作。

但基于 GWAS 的研究往往需要大量样本，单一数据源的数据量很难满足一项研究所需的足够样本量。多机构的数据融合是最佳解决方案，既能提高样本量，又能扩充样本维度，提升研究质量，同时最大程度地利用了数据。但在具体实践中，跨机构基因数据协作和共享存在包括隐私安全和合规性等诸多问题。如何合理有效保护这些敏感信息，规避不必要的隐私泄露风险是广泛推行基因数据分享和联合分析、实现生物医疗数据融合所面临的主要挑战之一。

为应对这一挑战，锘崴科技引入英特尔® SGX 来构建基于硬件的可信执行环境（TEE），并在其上打造可进行隐私保护计算的 iPRIVATES 框架方案。新框架能通过融合不同的隐私保护计算技术，来满足用户在不同 GWAS 研究场景下对数据隐私保护的需求，为医疗数据“可用不可见”赋能。

■ 案例描述与成效

由锘崴科技打造、带有隐私保护的GWAS技术框架 iPRIVATES，能实现在不分享明文数据（个体基因数据）的基础上，支持多种疾病的 GWAS 研究，为解决生物医学数据的共享问题提供了新思路。

iPRIVATES 框架融合了多种面向 GWAS 分析的技术和算法，例如可定制的基因组数据预处理模块、基于主成分分析（Principal component analysis，PCA）的人口分层模型、基于逻辑回归（LR）和似然比（Likelihood Ratio，LLR）检验的关联分析模型。上述设计能灵活地集成和配置不同的 GWAS，方便识别 SNPs 与许多不同类型的特征（如某些重大疾病）之间的关联。

但其在模型评估阶段涉及许多敏感信息，如模型参数、模型输入数据、模型结果（例如匹配结果）等无法由传统联邦学习方法提供隐私保护，可能会出现泄露。此外，内部攻击也是方案中的中心节点（Global Service Provider, GSP）面临的威胁之一。例如在建立逻辑回归模型时，中间统计数据可能会泄露敏感信息。

为此，锘崴科技与英特尔合作，通过融合英特尔® SGX 来构建更为安全的数据共享方法和流程。基于 iPRIVATES 框架的系统，如图 2-6-11 所示，来自各个医院的数据可通过客户端汇集到锘崴信隐私保护计算平台进行处理分析，锘崴科技融合英特尔® SGX，通过软硬件结合的方式在底层构建可信执行环境（TEE），以确保基因数据共享过程中每一环节的安全，不仅实现了不分享明文个体数据，同时也对模型本身进行保护。

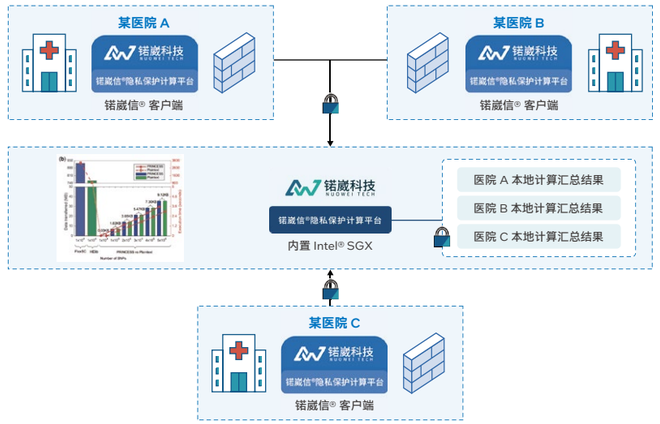


图 2-6-11 iPRIVATES 框架示意图

其中，框架中用到的安全联邦学习（Secure Federated Learning, SFL）技术，是锘崴团队在传统联邦学习技术基础上提出的创新技术，其能够有效消除传统联邦学习中存在的信息泄露问题。安全联邦学习通过软硬件结合的方式，仅分享经过加密的中间统计值，不分享明文个体数据，同时也对模型本身进行保护，保证数据共享的全链路隐私安全，兼顾隐私保护和跨机构数据共享的双重目标。

在多家医疗机构开展的面向隐私保护计算的 GWAS 研究中，iPRIVATES 框架在计算精度、算法时间方面都等价于数据物理集中的方式，同时其产生的研究结果，即特征靶点也与集中式计算结果一致。但在计算效率上，iPRIVATES 框架远优于传统计算方式，这意味着类似方法及理念在解决生物、医疗多中心数据协作方面，有着巨大的可行性和潜力。

运用 OpenFL 推动联邦学习方案落地医疗领域

■ 案例背景

随着联邦学习方法在更多隐私保护计算场景中获得应用，如何提升这一方案的可用性，使其与更多 AI 方法、框架和工具实施协同，更便捷有效地实现落地，也是包括英特尔在内的一系列前沿厂商所关注的问题。

由英特尔开源的 OpenFL（开放联邦学习，github.com/intel/openfl）是一个基于 python 的机器学习框架，其可以与 TensorFlow 和 PyTorch 构建的训练管道（pipeline）配合使用。OpenFL 秉承联邦学习的思路，允许开发者在远端数据所有者（即合作者）的节点上训练机器学习 / 深度学习模型。由于模型是在合作者节点的硬件上训练的，因此训练模型的数据也不会被移动或复制，只有模型的权重更新和参数会分享给模型所有者，从而保证了数据具有“可用不可用”的安全特性。

如图 2-6-12 所示，与传统联邦学习流程一致，OpenFL 架构中参与协作的合作者（Collaborator）都需要导入预设的联邦学习计划、机器学习 / 深度学习模型代码，以及本地数据集，各个节点之间的协调和执行是由各节点间共享的联邦学习计划定义。此外，计划还会定义联邦学习流程中的各项设置，如 IP 地址，训练中的 Batch 大小以及训练轮次等。在启动联邦学习之前，使用者可使用 OpenFL 的命令行界面（CLI），手动为每个参与者共享联邦学习计划和模型代码。

当联邦学习启动后，OpenFL 后端允许合作者通过远程调用的方式向聚合服务器（Aggregator）发送请求，询问接下来应该执行哪个任务（如启动某个机器学习模型的训练）。借助这种方式，聚合服务器可动态地选择将具体任务分配给每个合作者。当合作者完成当前任务后，其会将更新的模型权重（以及汇总的参数，如模型精度和本地数据集大小等）上报给聚合器。聚合服务器会将更新信息合并成一个统一共识模型（global consensus model），然后合作者再从聚合器服务中检索新的统一共识模型的权重，进行新一轮的任务，直至训练任务完成。

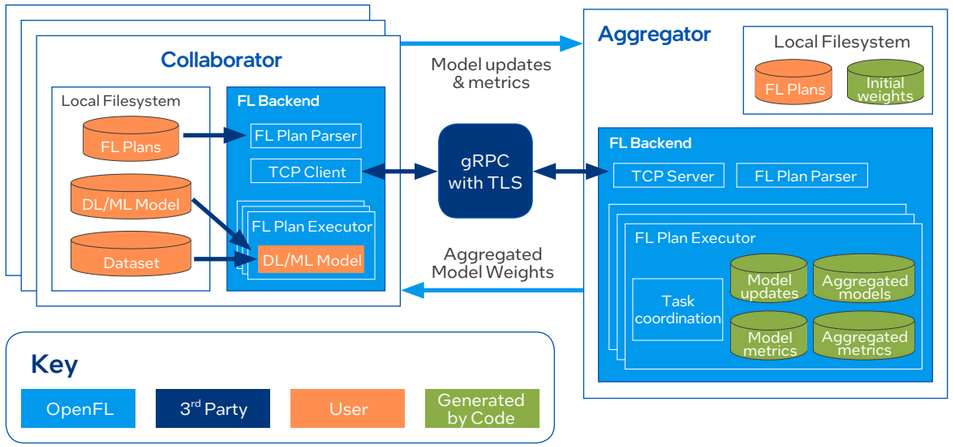


图 2-6-12 OpenFL 架构及工作流程

在解决了多数据源的协作模式之后，OpenFL 中各类机器学习 / 深度学习模型就可以在分布式的环境中使用不同数据集开展训练。而对于关键的数据安全性问题，OpenFL 会通过各类安全设计，包括引入硬件可信执行环境（TEE）等方式来予以解决。这其中，OpenFL 架构中对英特尔® SGX 有着良好的支持。在 OpenFL 的工作流程中，英特尔® SGX 能够通过内存中的“飞地”对数据和模型 IP 提供有效保护。更多英特尔® SGX 工作方法，请参阅第 67 页“英特尔® 软件防护扩展”部分所述。

■ 案例描述与成效

作为便捷可用的联邦学习落地方案，目前 OpenFL 正在全球各地的联邦学习方案的落地部署中获得重视，在医疗领域也同样如此。近年来，医疗领域运用 OpenFL 开展了一系列卓有成效的联邦学习方案落地，这里以辐射对宇航员的生理影响的评估为例：

来自 NASA 前沿发展实验室（Frontier Development Lab，FDL）的科学家们正借助联邦学习方法来研究宇航员的健康，从而更好地了解空间辐射对人类的生理影响。由于啮齿动物的辐射数据可作为人类辐射数据的同源物，因此 FDL 的科学家们利用 OpenFL 框架提出了一个创新的病灶生物标志物检测算法。该算法利用辐射对小鼠的影响数据，来训练面向人类的模型，这个模型将更准确地预测受到辐射影响的基因，与免疫反应的相关性。

研究中，借助 OpenFL 框架，部署在美国国家航空航天局、梅奥诊所和美国国家航空航天局基因实验室等机构的 CRISP 2.0 模型（因果关系和推理搜索平台）得以实现联合训练，而无需将数据转移 / 共享到某个集中位置。这一点至关重要，原因在于，一方面每个机构的数据都是私有，且具有隐私风险；另一方面，在航天器上传输大量数据可能会带来高昂的成本。

而通过 OpenFL，研究人员能用一个因果推理方法集合（预先在小鼠数据上训练得到）去初始化联合实验，并在各个合作者提供的数据集中，选择最高方差的人类基因和各自的小鼠同源物，进行 30 多轮的联合训练，最后使用 CRISP 2.0 输出结果，并进行进一步分析和洞察。通过对前 50 个具有强共性特征的分析，研究人员发现了以前未识别的基因 SLC8A3，并将其作为进一步研究的潜在因果目标。

小结

作为医疗 AI 应用发展的重要“燃料”，更多高质量医疗数据无疑可以有力提升 AI 应用的性能，但如何解决其中的数据安全和隐私保护问题，一直是医疗行业推动 AI 发展时需要面对的挑战。而联邦学习方法，已被证明是应对这一挑战的良好方案。

现在，英特尔正与众多医疗、科研机构展开合作，借助英特尔® SGX 以及基于英特尔® 架构的处理器等先进软硬件产品，使联邦学习方法在保证数据安全可信的情况下，有效解决医疗机构 AI 训练中面临的训练数据匮乏问题，进一步推动医疗 AI 应用的快速发展。

AI 技术加速蛋白质结构预测

AlphaFold2 实现蛋白质结构预测加速

蛋白质结构预测的价值

作为生物体中最重要的组成部分之一，蛋白质的结构揭示着生命中的许多本质问题。因此，在生物学、医学、药学乃至农业、畜牧业等领域中，通过对蛋白质三维结构的有效解析与预测，发现其中脱氧核糖核酸（DNA）、核糖核酸（RNA）以及蛋白质（包括多肽[62]、氨基酸）之间的“转录 - 翻译”关系，并清晰呈现生物体内的信息传递路径，一直是相关领域科研机构、实验室和企业开展探索，对生物体运行和变化的规律实施更深层次的诠释，进而推动各类技术创新和产品研发的重要方法。

这些解析与预测工作传统上通常依赖基于实验方法的蛋白质结构解析工具，包括 X- 射线晶体衍射、冷冻电镜、核磁共振等来完成。但这些方法的效率已逐渐赶不上氨基酸序列的增加速度，后果之一便是海量待测样品 / 序列可能会在实验室中等待数月乃至数年才能得到解析。以 UniProtKB/Swiss-Prot 数据库搜集和整理的数据为例，单从实验获得的已知蛋白序列就已高达 57 万条之多[63]。

AI 技术的高速发展正为破解上述效率问题带来新的思路。人们开始将 AI 中的深度学习等方法运用于蛋白质结构预测，例如经典的 ResNet 网络就曾被用来开展高水平的蛋白质结构预测[64]。而今天，由 DeepMind 在 2020 年 CASP 14[65] 上提出的 AlphaFold2 方案尤其令人瞩目，它以惊人的 92.4 分（GDT_TS 分数）的表现实现了原子级别的预测精度，被认为“已可替代传统实验方法”[66]。

基于 AlphaFold2 的蛋白质结构预测方法[67]

得益于全新的设计思路，AlphaFold2 为人们提供了完整的端到端蛋白质三维结构预测流程。如图 2-7-1 所示，其工作流程大致可分为预处理（Preprocessing）、深度学习模型推理（DL Model Inference）以及后处理（Postprocessing）三个阶段，各阶段执行的功能如下：

- **预处理**：由于初始输入的氨基酸序列所含信息往往较少，因此 AlphaFold2 在预处理阶段会先利用已知信息（包括蛋白质序列、结构模板）来提升预测精度。包括借助一些蛋白质搜索工具在特定序列数据库中使用多序列比对（Multiple sequence alignment，MSA）方法，以及在特定结构数据库中进行模板搜索，从而获得不同蛋白质之间的共有进化信息；
- **深度学习模型推理**：在该阶段中，AlphaFold2 首先会借助嵌入（Embedding）过程，将来自预处理阶段的模板 MSA 信息、MSA 和目标构成 MSA 表征（MSA representation）的三维张量，同时也将模板邻接信息和额外的 MSA 构成邻接表征（pair representation）的三维张量，随后两种表征信息会通过一个由 48 个块（Block）组成的 Evoformer 网络进行表征融合。在这一进程中，模型将通过一种 Self-Attention 机制来学习蛋白质的三角几何约束信息，并让两种表征信息相互影响来使模型推理出相应的三维结构，且循环三次；
- **后处理**：这一阶段，AlphaFold2 将使用 Amber 力场分析方法，对获得的三维结构参数优化，并输出最终的蛋白质三维结构。

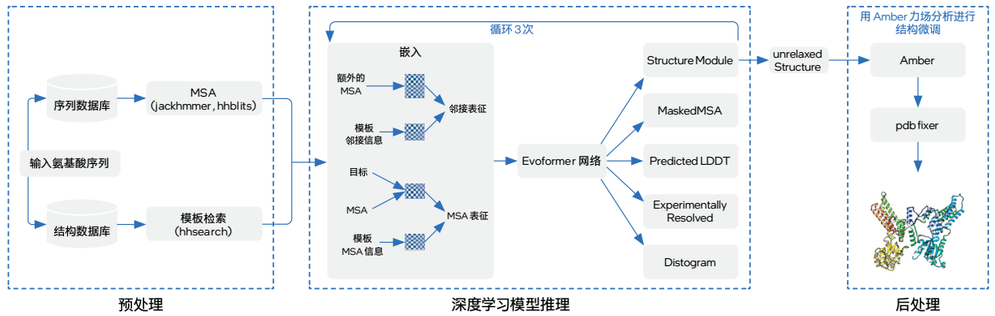


图 2-7-1 AlphaFold2 基本架构

[62] 肽是α - 氨基酸以肽键连接在一起而形成的化合物，是蛋白质水解的中间产物，由三个或三个以上氨基酸分子组成的肽叫多肽。
[63] 数据援引自 UniProtKB/Swiss-Prot 数据库官网：https://web.expasy.org/docs/relnotes/relstat.html。
[64] 信息援引自《Improved protein structure prediction by deep learning irrespective of co-evolution information》，Jinbo Xu, Matthew McPartlon & Jin Li，https://www.nature.com/articles/s42256-021-00348-5
[65] CASP，即结构预测的关键评估竞赛（Critical Assessment of Structure Prediction），于 1994 年启动，是对蛋白质结构的计算预测进行基准测试的一种手段。DeepMind 在 2020 年的 CASP 14 上提出了 AlphaFold2 算法。
[66] 一般认为，AI 方法的预测精度（GDT_TS 分数）超过 90 分，可认为预测结果与实验方法得到的蛋白质结构基本一致。
[67] 本节中有关基于 CNN 及 M-CNN 的 HCS 的技术描述，详情请参阅：Godinez et al, A multi-scale convolutional neural network for phenotyping high-content cellular images. Bioinformatics, 2017

软硬件配置建议

对于利用 AlphaFold2 来开展蛋白质结构预测，可以参考以下基于英特尔® 架构平台的软硬件配置，来进行系统部署。

名称	规格
处理器	第四代英特尔® 至强® 可扩展处理器，或第三代英特尔® 至强® 可扩展处理器（例如英特尔® 至强® 铂金 8358 处理器）及以上
超线程	ON
睿频加速	ON
内存	16 x 32GB DDR4 3200MHz 及以上
存储	英特尔® 固态盘 S4510 系列及以上
操作系统	CentOS Linux 8 或最新版本
Linux 核心	4.18.0-240.22.1.el8_3.x86_64 或最新版本
Python 版本	基于英特尔® 架构优化的 Python 3.9.7 或最新版本
PyTorch 版本	IPEX-2.0.100+ cpu 或更高版本
JAX 版本	0.3.14 或最新版本

基于英特尔® 至强® 可扩展处理器平台开展 AlphaFold2 优化

各行业和领域内的使用者在借助 AlphaFold2 进行蛋白质结构预测时所面临的重大挑战之一，就是如何保证有充沛的算力去应对预测各环节中所需庞大的计算量，尤其随着预测蛋白质序列长度不断加长，计算复杂度也正变得越来越大。为此，使用者需要更加充分地挖掘硬件的计算潜力来提升执行效率；以及为缩短结构预测时间而利用更多计算节点，来构建效率更高的并行计算方案等。

英特尔为这一工作提供了从算力平台、AI 加速能力到软件优化的全面支持。借助英特尔® 至强® 可扩展平台提供的内置 AI 加速能力，对运算和存储性能的均衡设计，以及对硬件和软件协同优化能力的兼顾，英特尔为 AlphaFold2 全流程提供了端到端的全面优化。针对 AlphaFold2 的设计特点，优化方案主要聚焦在预处理和模型推理两个层面，推出了 9 项优化措施。这些优化项可以分别作用于第四代或第三代英特尔® 至强® 可扩展平台上。

优化项	第四代英特尔® 至强® 可扩展平台	第三代英特尔® 至强® 可扩展平台
高通量优化	√	√
深度学习模型迁移至面向英特尔® 架构优化的 PyTorch	√	√
引入PyTorch JIT	√	√
切分 Attention 模块和算子融合	√	√
挖掘多核心优势	√	√
借助 TPP 技术降低推理过程中的内存消耗	√	
提供对 DDR5 内存与大容量缓存的支持	√	
引入英特尔® AMX_BF16 在保证精度的前提下加速推理过程	√	
采用高带宽内存 HBM2e 增加访存通量	√	

表 2 基于英特尔® 至强® 可扩展处理器的优化

预处理阶段 - 高通量优化

预处理阶段的高通量计算需求，使 AlphaFold2 在执行时面临巨大的并行计算压力。借助第四代或第三代英特尔® 至强® 可扩展处理器的多核优势，及内置的英特尔® AVX-512 技术，方案能实现针对预处理阶段的高通量优化。

如前所述，AlphaFold2 会在预处理阶段对特定序列数据库和结构数据库中的已知序列 / 模板信息进行搜索，包括使用 jackhmmer 等蛋白质搜索工具来执行 MSA 方法，即从数据库中抽取和输入与氨基酸序列相近的序列并进行对齐，其目的是找出同源的序列 / 模板组成表征信息，来为后续推理过程提供输入，由此提高预测精度。

这一过程需要执行大量的向量 / 矩阵运算。以模板搜索为例，其本质为计算两个隐马尔可夫模型（Hidden Markov Model，HMM）间的距离。当输入的氨基酸序列很长（例如执行中输入长度达数百的氨基酸序列）且需并行执行大量实例时，如果无法让处理器的算力 “火力全开” 去提升平台的并行计算效率性能，那么整个预处理过程的效率就会变得乏善可陈。

在优化方案中，一方面英特尔® 至强® 可扩展处理器出色的微架构设计，尤其是多核心、多线程和大容量高速缓存，可以保证 AlphaFold2 获得充足的总体算力，满足整个结构预测过程所需；另一方面，内置的英特尔® AVX-512 也为方案提供了更进一步的性能调优空间。针对序列 / 模板搜索所需的大量向量 / 矩阵运算需求，英特尔® AVX-512 能以显著的高位宽优势（最大可提供 512 位向量计算能力），来提升计算过程中的向量化并行程度，有效提升向量 / 矩阵运算效率。

实战中，使用者在预处理阶段可以参考以下代码示例进行调优（以下代码示例以第四代英特尔® 至强® 可扩展处理器为例）。在指令调用优化设定阶段，代码示例如下：

```
//引入头文件，以 jackhmmer 热点函数 calc_band 为例
#include <immintrin.h>
#include <tbb/tbb.h>
```

```
//AVX512 矩阵初始化
register const __m512i masks[] = {
  _mm512_set_epi64(m64，m64，0x0，0x0，0x0，0x0，0x0，0x0),
  _mm512_set_epi64(0x0，0x0，  m64，  m64，0x0，0x0，0x8，0x0),
  _mm512_set_epi64(0x0，0x0，  0x0，  0x0，m64，m64，0x0，0x0),
  _mm512_set_epi64(0x0，0x0，  0x0，  0x0，0x0，0x0，m64，m64)
  };
```

```
//AVX512 逻辑运算
data =mm512_bslli_epi128(sv[sv_index]，1);
  data = mm512 or epi64(data，beginvx4);
  data = mm512 and epi64(data，  masks[maskindex]);
  sv[sv_index] =_mm512_and_epi64(sv[sv_index]，inverse_masks[mask_index]);
  sv[sv_index] =_mm512_or_epi64(sv[sv_index],data);
  }
```

```
//AVX512 数值运算
sv[k] =_mm512_subs_epi8(sv[k]，*rsc++);
xEv4=_mm512 max_epu8(xEv4，sv[k]);
```

在使用英特尔® ICC 编译器进行代码优化编译设定阶段，代码示例如下：

```
#以 hh-suite 为例，cmake 设定优化编译
cd hh-suite
mkdir build && cd build
cmake \
  -DCMAKE_INSTALL_PREFIX=`pwd`/release \
  -DCMAKE_CXX_COMPILER="icc" \
  -DCMAKE_CXX_FLAGS_RELEASE="-O3 -march=icelake-server" \
  ..
make && make install
release/bin/hhblits -h
export PATH=`pwd`/release/bin:$PATH
```

```
#以 Hmmer 为例，configure 设定优化编译
source <intel-oneapi>/tbb/latest env/vars.sh
cd hmmer-3.3.2/hmmer
make clean
git clone https://github.com/EddyRivasLab/easel.git
cd easel && make clean && autoconf
./configure --prefix=`pwd` && cd ..
CC=icc CFLAGS="-O3 -march=icelake-server" \
./configure --prefix=`pwd`/release
make && make install
release/bin/iackhmmer -h
export PATH=`pwd`/release/bin:$PATH
```

在预处理的 MSA 并行计算优化设定阶段，代码示例如下：

```
#引入库
from multiprocessing import Process,Manager
```

```
#并行的模块放入多个函数实现
if self.run_in_parallel:
  def uniref90 search(input fasta path, return dict):
  # do something
  return_dict['uniref90_msa'] = jackhmmer_uniref90_result

  def mgnify_search(input_fasta_path, return_dict):
  # do something
  return dict['mgnify_msa'] =jackhmmermgnify_result

  def bfd_search(input_fasta_path, return_dict):
  # do something
  return dict['bfd msa'] = bfd _msa
```

```
#开启多个子进程
mgmt = Manager()
return_dict = mgmt.dict()
p_uniref90 = Process(target=uniref90_search,args=(input_fasta_path,return dict,))
p_mgnify = Process(target=mgnify_search,args=(input_fasta_path,return_dict,))
p_bfd = Process(target=bfd search,args=(input_fasta_path,return_dict,))
p_uniref90.start()
p_mgnify.start()
p_bfd.start()
p_uniref90.join()
p_mgnify.join()
p_bfd.join()
```

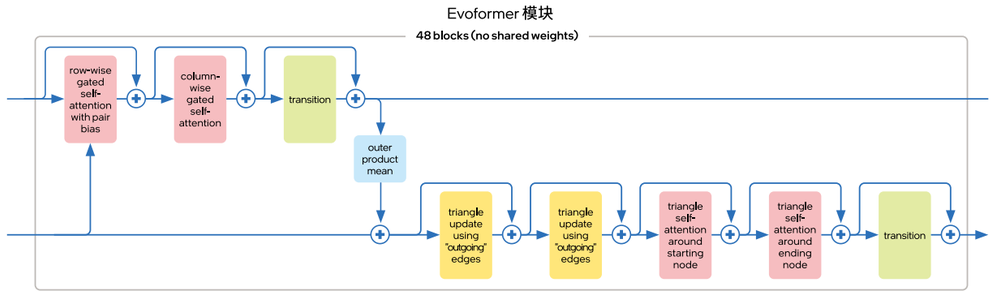
```
#收集结果
jackhmmer_uniref90_result = return_dict['uniref90_msa']
jackhmmer_mgnify_result = return_dict['mgnify_msa']
bfd_msa = return_dict['bfd msa']
```

模型推理阶段 - 深度学习模型迁移至面向英特尔® 架构优化的 PyTorch

原始版本的 AlphaFold2 是基于 DeepMind 的 JAX 和 haiku-API 做的网络实现，但目前 JAX 上还没有面向英特尔® 架构平台的优化工具。而 PyTorch 拥有良好的动态图纠错方法，与 haiku-API 有着相似的风格，并可以采用面向 PyTorch 的英特尔® 扩展优化框架（Intel® Extensions for PyTorch，IPEX，可由英特尔® oneAPI AI 工具套件提供）。为实现更好的优化效果，方案选择将深度学习模型迁移至面向英特尔® 架构优化的 PyTorch，并最终逐模块地从 JAX/haiku 上完成了代码迁移。

模型推理阶段 - 引入 PyTorch JIT

为提高模型的推理速度，便于利用 IPEX 的算子融合等加速手段，优化方案中还对迁移后的代码进行了一系列的 API 改造，在不改变网络拓扑的前提下，引入 PyTorch Just-In-Time (JIT) 图编译技术，将网络最终转化为静态图。



Name	Self CPU %	CPU total %
aten::einsum	0.07%	42.23%
aten::bmm	26.73%	37.51%
quantized::Linear_dynamic	20.15%	20.16%
aten::clone	0.08%	16.45%
aten::copy_	16.35%	16.35%
aten::add	12.62%	12.62%
aten::softmax	0.00%	8.76%
aten::_softmax	8.76%	8.76%

Large Tenors (~140MB per tensor)

图 2-7-2 Evoformer 模块的热点算子

模型推理阶段 - 切分 Attention 模块和算子融合

AlphaFold2 的嵌入过程是构成 MSA 表征张量和邻接表征张量来作为 Evoformer 网络输入的关键步骤。从其算法设计可以获知，其注意力模块（attention unit）中包含了大量的偏移量（bias）计算。

这种偏移量计算是通过张量间的矩阵运算来完成的，运算过程中会伴随张量的扩张。当张量达到一定规模后，扩张过程对内存容量的需求就会变得巨大。以一个 “5120 x 1 x 1 x 64” 的张量为例，其初始内存需求为 1.25MB，但在扩张过程中，对内存容量的需求却可达 930MB。

这就使 AlphaFold2 在嵌入过程中面临两个问题，一方面是巨大的内存峰值压力，其需求量会使内存资源在短时间耗尽，尤其是内存峰值在相互叠加之后，可能造成推理任务的失败；另一方面，大张量运算所需的海量内存也会带来不可忽略的内存分配过程，从而增加执行耗时。

为此，英特尔提出了 “对注意力模块进行大张量切分（tensor slicing）” 的优化思路，即将大张量切分为多个较小的张量，来降低扩张中的内存需求。例如，将上述 “5120 x 1 x 1 x 64” 的张量切分为 “320 x 1 x 1 x 64” 后，其扩张所需的内存就由 930MB 降至 59.69MB，仅为未进行张量切分时的 6.4% 左右，有效消减了内存峰值压力。相关代码示例如下：

```
def _slice_attention(self,q_data,m_data,bias,nonbatched_bias):
  ### avoiding huge memory cost
  ### threhold is ajustable
  threhold = 1000
  unit 320                                  #unit is ajustable
  if q_data.size()[0] > threhold:     #and is_stop_slice == 0:
    res = self.ones_like(q_data)
    for i in range(q_data.size()[0]//unit):
      q_sub_data = q_data[unit*i:unit*(i+1)]
      m_sub_data = m_data[unit*i:unit*(i+1)]
      bias_sub = bias[0:unit]
      res[unit*i:unit*(1+1)] = self.attention(q_sub_data,m_sub_data,bias_sub,nonbatched_bias)
    return res
  else:
    return self.attention(q_data,m_data,bias,nonbatched_bias)
```

此外，英特尔发现，利用 PyTorch 自带的 Profiler 对 AlphaFold2 的 Evoformer 网络进行算子跟踪分析时， Einsum 和 Add 这两种算子占用了大部分的算力资源。因此，英特尔就考虑使用 IPEX（建议版本为 IPEX-2.0.100+ cpu 或更高）提供的算子融合能力，来实现上述两种计算过程的融合。

传统深度学习计算过程都是逐一操作，例如 Einsum 计算过程结束后，函数返回值需要在 Python 进程中建立一个临时缓存，然后通过调用 Add 算子，再次进入 oneDNN 完成第二个函数的运算，这中间来回折返的过程时间消耗不可忽略。如图 2-7-3 所示，算子融合带来的优势就在于，在前一操作结束后可以马上执行后一操作，节省了中间建立临时缓存数据结构的时间。同时，从时间轴上不难看出，经过融合后，两个连续的算子合并为一个，用时也显著缩短。

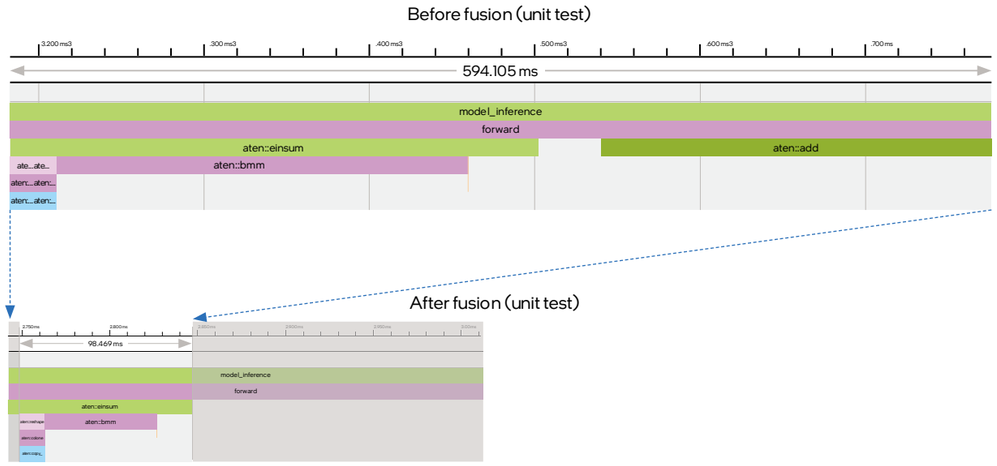


图 2-7-3 算子 Einsum+Add 融合效果图

模型推理阶段 - 挖掘多核心优势

为了让推理性能在多实例进程中获得更接近线性的增长表现，优化方案还借助英特尔® 至强® 可扩展平台提供的高效且更为均衡的计算和存储优势，实施了有针对性优化。

首先，借助基于 NUMA 架构的核心绑定技术，来充分挖掘至强® 可扩展处理器的多核优势。得益于英特尔® 至强® 可扩展处理器在微架构设计上的优势，物理核与物理核之间的数据通信平均延时较短，每个 NUMA 在并行计算中的工作效率也会更高。如图 2-7-4 所示，这一技术可对处理器节点以及访问本地内存进程予以精确控制，让每个推理工作负载都能稳定地在同一组核心上执行，并优先访问对应的近端内存，从而提供更优也更稳定的并行算力输出。在执行中可使用以下 numactl 指令：

```
numactl -C $core_ids -m $socket_id $command
```

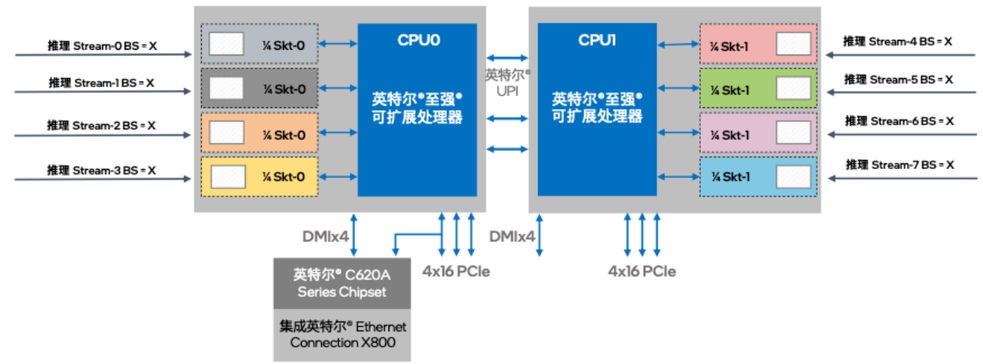


图 2-7-4 英特尔® 至强® 可扩展处理器提供多核并行算力输出

模型推理阶段 - 借助 TPP 技术降低推理过程中的内存消耗

在深度学习系统开发中，诸如算子（Operators）、算法概念（Algorithmic Concepts）以及计算模式（Computational Motifs）等编程范式（Programming Paradigm）通常会面向特定平台进行调优，这会对系统的构建便利性、性能调优以及可移植性造成障碍。为此，张量计算原语（Tensor Processing Primitives，TPP）技术是在 2D 张量上定义了一组低层级的基本算子，通过有效且可移植的张量级算子来应对这一问题。TPP 可被看成是一种虚拟的张量指令集架构，能将英特尔® AVX-512 等物理指令集予以抽象，并生成经优化的平台代码。

根据自身软硬件特性，英特尔面向 PyTorch 对 TPP 进行了扩展。面向 PyTorch 的英特尔® TPP 扩展（Intel® Tensor

Processing Primitives Extension for PyTorch）不仅能让开发者直接使用 TPP 调用英特尔® oneAPI 等库来生成优化代码，也可利用面向 PyTorch 的 TPP 作为构建块，来表示底层张量计算。

引入 TPP 技术能让 AlphaFold2 在通用矩阵乘法（GEMM）等计算中获得优势，降低内存消耗并更好地利用第四代英特尔® 至强® 可扩展处理器所具备的大容量末级缓存优势，有助于加速诸如在 Evoformer 模块中需要进行大量的狭长矩阵乘法等运算。对于在处理器上执行的矩阵乘法计算，一般会采用两种重要的优化方式：

- 以单指令多数据（SIMD）方式处理数据；
- 优化内存访问模式，提升缓存命中率来提高数值计算和访存效率。

通过引入面向 PyTorch 的英特尔® TPP 扩展，英特尔在 AlphaFold2 实现了以上两种优化。如图 2-7-5 所示，一方面由 libxsmm（小矩阵乘法函数库）构建起来的 TPP BRGEMM（Batch Reduce General Matrix Multiplication）能最大化利用第四代英特尔® 至强® 可扩展处理所内置的 SIMD 运算单元，同时小矩阵乘法也能有效提高缓存命中率，使处理器的大容量末级缓存优势在计算过程中获得更充分的利用。

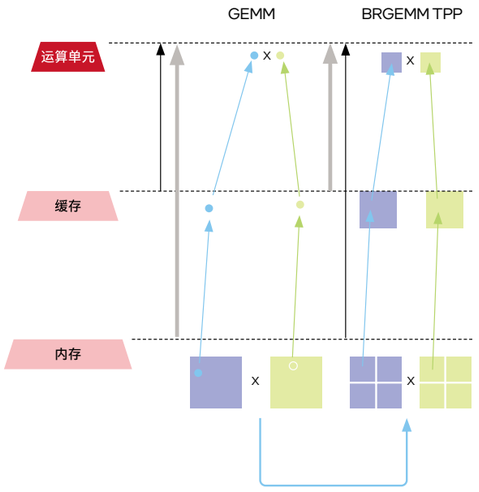


图 2-7-5 以 TPP 技术来充分利用处理器的缓存优势

同时，TPP 技术的引入，令狭长矩阵乘法的空间复杂度从 O(n^2) 降为 O(n)，这使得运算过程中所需的内存峰值大幅降低，有效缓解长序列蛋白质结构预测工作中面临的“序列长度天花板”问题。如图 2-7-6 所示，在一项对比测试中，随着所预测蛋白质序列长度的增加，使用 TPP 技术的测试组（橙色线）所需内存峰值为线性增加，而未使用 TPP 技术的测试组（灰色线）所需内存峰值呈现指数增加状态，很快就攀升至 TB 级（数据显示，当人们对 LRP2 蛋白进行结构预测时，其 4700aa 的序列长度要求的内存容量就远大于 1.3TB），形成阻碍应用工作效能发挥的“峰值内存墙”。

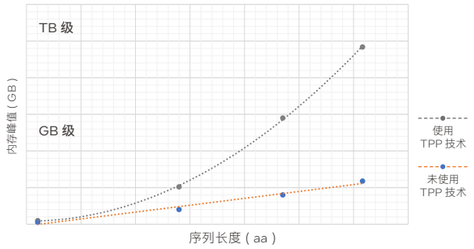


图 2-7-6 TPP 技术带来所需内存峰值的大幅降低

实战中，使用者可以参考以下代码示例来构建 TPP BRGEMM，并替换原始的自注意力模块。

```
//BRGEMM 优化代码
// auto q = at::mul(at::einsum("bqa,ahc->bqhc", {q_data, query_w}), (1.0/sqrt(key_dim)));
{
RECORD_SCOPE(alpha_q_gemm, {q, q_data, query_w});
  {
    qkv_vnni_trans_tpp(&query_w_a[0][0][0], &qkv_w_vnni_a[0][0][0]);
    RECORD_FUNCTION("parallel for", std::vector<c10::IValue>());
    #pragma omp parallel for collapse(2)
    for(int i=0; i < B_t; i++){
      for(int j=0; j < s_t; j += QKV_BLOCKSIZE){
        float tmp[QKV_BLOCKSIZE][N_t][H_t];
        zero_tpp(&tmp[0][0][0]);
        q_brgemm tpp(&q_data_a[i][j][0], &qkv_w_vnni_a[0][0][0], &tmp[0][0][0], 1);
        scale_tpp(&tmp[0][0][0], &tmp[0][0][0], alpha); q_convert_tpp(&tmp[0][0][0],
&q_a[i][j][0][0]);
      }
    }
  }
}
```

```
#替换原始的自注意力模块
try:
  from alphafold.pytorch_jit.basics import GatingAttention
  from pcl_pytorch_extension.alphafold.Alpha Attention import GatingAttentionOpti_forward
  GatingAttention.forward = GatingAttentionOpti_forward
  is_pcl = True
except:
  is_pcl = False
  print('[warning] No PCL extension detected, will fallback to JIT mode')
```

模型推理阶段 - 提供对 DDR5 内存与大容量缓存的支持

通过对算法架构的解析可知，AlphaFold2 中大量的矩阵运算过程都需要内存予以支撑，因此内存性能是影响 AlphaFold2 性能的重要因素。而随着预测序列长度的增加，计算中所需的内存也会成倍增加，内存性能，尤其是内存带宽对系统整体性能的影响也会更为明显。

与此同时，更优的缓存策略也能让 AlphaFold2 进一步发挥潜能。由于张量间的矩阵运算会涉及大量的内存数据访存，而更靠近处理器运算单元末级缓存在延迟性能上比内存高出一个数量级。因此在复杂的矩阵运算中，更多的热数据通过末级缓存访存而非内存可以带来显著的性能提升。

第四代英特尔® 至强® 可扩展处理器对 DDR5 内存的支持，以及所具备的大容量末级缓存，为张量吞吐量的提升提供了更佳途径。新一代 DDR5 内存不仅频率更高、工作电压更低，还具有远超 DDR4 内存的带宽速度。与 DDR4 内存 25.6GBps（3,200MHz）的带宽相比，DDR5 内存带宽达到了 38.4GBps（4,800MHz）以上，提升幅度超过了 50%。同时，新处理器的末级缓存也由上一代的最高 60MB 提升至本代的最高 112.5MB，提升幅度达 87.5%[68]。性能更高的内存与容量更大的末级缓存，使 AlphaFold2 推理过程中关键的张量吞吐获得了显著提升。

模型推理阶段 - 引入英特尔® AMX_BF16 在保证精度的前提下加速推理

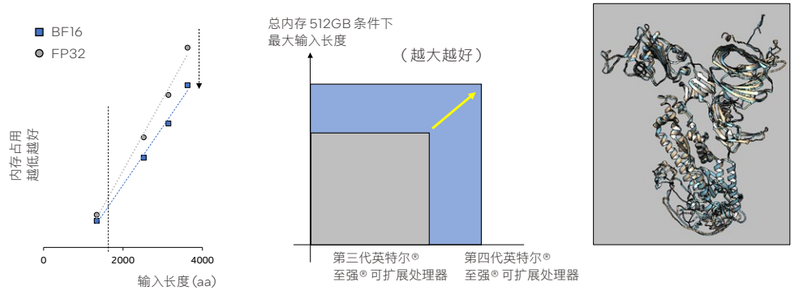
第四代英特尔® 至强® 可扩展处理器面向深度学习应用推出的“杀手锏”之一就是创新的 AI 加速引擎，即英特尔® AMX。作为矩阵相关的加速器，英特尔® AMX 能显著加速基于 CPU 平台的深度学习推理和训练，提升 AI 整体性能。英特尔® AMX 对 INT8、BF16 等低精度数据类型都有着良好的支持（通过 AMX_INT8、AMX_BF16 等不同指令集执行操作），如图 2-7-7 最右侧所示，来自 AlphaFold2 的实际预测结果表明，BF16 数据类型在精度上有着不逊于 FP32 数据类型的表现。

针对 AlphaFold2 推理过程所需的大量矩阵运算，AMX-BF16 能在保持较高精度的同时，提高计算速度并减少内存占用。如图 2-7-7 最左侧所示，在面向同一种蛋白质的结构预测工作中，BF16 占用内存明显低于 FP32，且这一趋势将随着所预测蛋白质序列长度的增加而愈发明显。究其原因，是因为英特尔® AMX 在解决矩阵乘法问题时，直接采用了分块矩阵乘法的方式。其内部所定义的 Tile 矩阵乘法（Tile Matrix Multiply Unit，TMUL）加速模块，能直接对矩阵寄存器中的数据实施矩阵运算操作，由此运算效率可得到大幅提升。实践数据表明，AlphaFold2 在推理过程中使用 AMX_BF16 后，推理效率可提升数倍之多。

而引入英特尔® AMX 带来的另一项优势，是使用者可以利用 AlphaFold2 开展更大序列蛋白质结构的预测。如图 2-7-7 中间所示，在总内存一致的情况下，基于第四代英特尔® 至强® 可扩展处理器的方案较第三代英特尔® 至强® 可扩展处理器有着更大的输入长度，可预测蛋白质序列更长。

模型推理阶段 - 采用高带宽内存 HBM2e 增加访存通量

与第四代英特尔® 至强® 可扩展处理器一同发布、采用了相同微架构的英特尔® 至强® CPU Max 系列中，还加入了对 HBM 的支持，这也能让运行在其上的 AlphaFold2 推理负载更进一步。作为一种采用 3D 堆叠技术的全新内存产品，HBM 能为



BF16 占用内存更低　　BF16 最大输入长度更大　　FP32 v.s. BF16 预测结构没有明显差异

图 2-7-7 不同精度数据类型在 AlphaFold2 中表现对比

[68] 具体产品细节可参阅英特尔官网相关英特尔® 至强® 可扩展处理器产品介绍：https://www.intel.cn/content/www/cn/zh/products/details/processors/xeon/scalable.html

AI 应用场景所需的各类计算负载提供更大的内存带宽支持。

- 每个英特尔® 至强® CPU Max 系列都拥有 4 个基于第二代增强型高带宽内存（HBM2e）的堆栈，总容量为 64GB（每个堆栈的容量为 16GB）；
- 由于能同时访问多个 DRAM 芯片，因此 HBM 在带宽方面相较 DDR 技术更具优势，其中 HBM2e 可提供高达 1TB/s 的带宽；
- HBM 内存可根据工作负载特性，以 “HBM Only”、“HBM Flat” 以及 “HBM Cache” 三种不同的模式，通过灵活的配置与 DDR5 内存一起协同工作。

在实践中，HBM2e 内存能有效缓解 AlphaFold2 推理负载中，大张量运算带来的海量内存需求，并以高带宽特性带来大幅访存通量提升，从而有力降低整体推理时长。在实战中，使用者可以参考以下代码示例来配置使用 HBM（Flat 模式）：

```
#对 HBM 内存进行配置
daxctl reconfigure-device -m system-ram dax0.0
daxctl reconfigure-device -m system-ram dax1.0
```

```
#对 NUMA 节点的内存绑定进行配置
numactl -m 1 ./a.out  # allocations must go to HBM (can't exceed 64GB)
numactl -p 1 ./a.out  # first allocations go to HBM, then to DDR (allows > 64GB)
```

英特尔优化方案在 AlphaFold2 上的实战

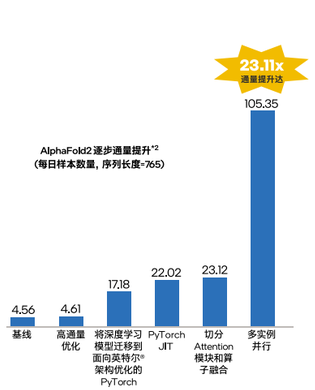
基于英特尔® 至强® 可扩展平台开展的 AlphaFold2 端到端优化，包括一系列并行计算能力优化举措的引入，使得整个 AlphaFold2 端到端处理过程的性能获得了质的提升，这在连续两代英特尔® 至强® 可扩展平台的实战中都获得了验证。

基于第三代英特尔® 至强® 可扩展处理器的 AlphaFold2 端到端优化，通量提升 23.11 倍

如图 2-7-8 所示，在基于第三代英特尔® 至强® 可扩展处理器的优化流程中，每个优化步骤获得的提升累积后，最后相比优化前通量提升可达 23.11 倍 [69]。

第四代英特尔® 至强® 可扩展处理器带来 AlphaFold2 通量再提升 3.02 倍

来自第四代英特尔® 至强® 可扩展处理器的优化加持，使



优化路线图	管线通量增幅	英特尔架构亲和性	英特尔 AI 软件
原始 AlphaFold2	基线		
第一步：高通量优化		英特尔® AVX-512	英特尔® 编译器
第二步：将深度学习模型迁移到面向英特尔® 架构优化的 PyTorch	+272.7%	英特尔® AVX-512	面向英特尔® 架构优化的 Python 英特尔® MKL
第三步：PyTorch JIT	+28.17%	英特尔® AVX-512	PyTorch with MKL
第四步：切分 Attention 模块和算子融合	+5.0%	英特尔® AVX-512	Intel® Extension for PyTorch (IPEX)
第五步：多实例并行	+355.67%	英特尔® 傲腾™ 持久内存，提供 TB 级内存支持	

图 2-7-8 基于第三代英特尔® 至强® 可扩展处理器的优化流程中多种优化措施带来的累计性能提升 [70]

[69] **测试配置：**

- **测试组：**处理器：2 x 英特尔® 至强® 铂金 8358 处理器，内存：16 x 32GB DDR4 3200MHz RDIMM + 16 x 256GB 英特尔® 傲腾™ 持久内存 200 系列 (Intel Optane NMB1XXD256GPSU4 DCPMM)，I/O 扩展：Raid Cntrlr - Trinity Dunes RAID Adapter,Intel RSP3TD160F，存储：Solidigm Youngsville Refresh SSDSC2KB038T801 S4510 Series，网络：SND I350-AM2 RJ45 Dual Port PCI-E4X_1KM，BIOS：Version: SE5C620.86B.01.01.0003.2104260124，Release Date: 04/26/2021，Linux 系统和 Kernel：Ubuntu 20.04 kernel-5.5.0-81-generic，Python 版本：基于英特尔® 架构优化的 Python 3.9.7，AI 框架：PyTorch 1.11.0+cpu, Intel PyTorch Extension 1.11.100 with oneDNN 2.6，其他工具和库：JAX 0.3.4, JAXlib 0.3.2+cuda11.cudnn82 ,HMMER 3.3.2, HH-Suite 3.3.0, OpenMM 7.5.1;
- **对比组：**处理器：2 x 英特尔® 至强® 铂金 8358 处理器，内存：32 x 128GB DDR4 3200MHz RDIMM，I/O 扩展：Raid Cntrlr - Trinity Dunes RAID Adapter, Intel RSP3TD160F，存储：Solidigm Youngsville Refresh SSDSC2KB038T801 S4510 Series，网络：SND I350-AM2 RJ45 Dual Port PCI-E4X_1KM，BIOS：Version: SE5C620.86B.01.01.0003.2104260124，Release Date: 04/26/2021，Linux 系统和 Kernel：Ubuntu 20.04 kernel-5.5.0-81-generic，Python 版本：基于英特尔® 架构优化的 Python 3.9.7，AI 框架：PyTorch 1.11.0+cpu, Intel PyTorch Extension 1.11.100 with oneDNN 2.6，其他工具和库：JAX 0.3.4 , JAXlib 0.3.2+cuda11.cudnn82 , HMMER 3.3.2, HH-Suite 3.3.0, OpenMM 7.5.1。

[70] 同脚注 69

AlphaFold2 的端到端通量获得进一步提升，如图 2-7-9 所示，与第三代英特尔® 至强® 可扩展处理器相比，融合 AMX_BF16、HBM 内存等技术的新平台能获得高达 3.02 倍的多实例通量提升 [71]。

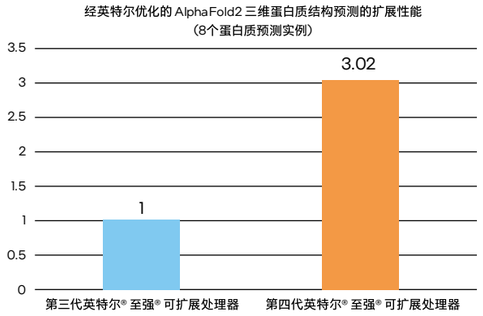


图 2-7-9 第四代英特尔® 至强® 可扩展处理器带来多实例通量提升 [72]

得益于性能强劲的算力表现和卓有成效的优化提升，如图 2-7-10 所示，已经有一系列不同序列长度的蛋白质已经基于第四代英特尔® 至强® 可扩展处理器进行了结构预测，并取得了令人满意的结果。

图 2-7-10 基于第四代英特尔® 至强® 可扩展处理器开展的蛋白质预测实例

[71] **测试配置：**

- **测试组：**处理器：2 x 英特尔® 至强® CPU MAX 系列 @ 1.90GHz，内存：128GB (8x16GB HBM2 3200MT/s，存储：1x 931.5G INTEL SSDPE2KX010T8，网络：1x Ethernet Controller X710 for 10GBASE-T，BIOS：SE5C7411.86B.8424.D03.2208100444，Linux 系统和 Kernel：CentOS Stream 8/5.19.0-rc6.0712.intel_next.1.x86_64+server，Python 版本：基于英特尔® 架构优化的 Python 3.9.7，AI 框架：PyTorch 1.11.0+cpu, Intel Extension for PyTorch 1.11.200 special branch for AlphaFold2，其他工具和库：JAX 0.3.14；
- **对比组：**处理器：2 x 英特尔® 至强® 铂金 8360Y 处理器 @ 2.40GHz，内存：512GB (16x32GB DDR4 3200MT/s)，存储：1x 894.3G INTEL SSDSC2KG96，网络：1x I210 Gigabit Network Connection, 2x Ethernet Controller 10G X550T，BIOS Version：WLYDCRB1.SYS.0021.P21.2106280839，Linux 系统和 Kernel：CentOS Linux 8/4.18.0-240.22.1.el8_3.x86_64，Python 版本：基于英特尔® 架构优化的 Python 3.9.7，AI 框架：PyTorch 1.11.0+cpu, Intel Extension for PyTorch 1.11.200 special branch forAlphaFold2，其他工具和库：JAX 0.3.14。

[72] 同脚注 71

在探索和验证上述端到端 AlphaFold2 优化方案、步骤和经验的过程中，英特尔也与同在寻求相关解决方案、专攻医药和生命科学研究和创新的产、学、研领域用户及合作伙伴们积极开展了广泛及深入的协作，这些协作起到了博采众长的效果，也为不断提升方案的普适性带来了助益。

同样，在优化方案基本定型，并展现了显著的通量提升效果，以及能够担起更长序列蛋白质结构预测重任的能力后，众多合作伙伴与用户也第一时间参考和借鉴了方案中的方法、经验与技巧，并结合自身特定的环境、应用现状和需求，开展了实战验证和更进一步的探索。

小结

凭借自身在蛋白质结构预测上的高可信度，以及远优于传统实验方法的效率和成本表现，AlphaFold2 正在 “AI for Science” 领域树起全新的里程碑。它不仅在生命科学领域掀起了颠覆式的革新，也成为了 AI 在生物学、医学和药学等领域落地的核心发力点。始终走在 AI 应用创新与落地一线的英特尔，也在这一过程中借助至强® 可扩展平台，包括其硬件层面的第三代英特尔® 至强® 可扩展处理器和第四代英特尔® 至强® 可扩展处理器，以及其软件层面的英特尔® oneAPI 工具套件等，基于这些软硬件之间的无缝组合与高效协作，以及多样化的 AI 优化方法，为 AlphaFold2 提供了端到端的高通量计算优化方案。

面向未来，英特尔还将继续携手科学前沿领域的合作伙伴，推进更多英特尔产品、技术与 AlphaFold2 等新技术开展交互与融合，在更多层面助力和加速 “AI for Science” 技术创新，让 AI 应用为各类前沿科学研究和探索带来更多加速、助力与收获。

英特尔架构产品与技术为医疗大模型加速

大模型技术为智慧医疗提供新动能

更具优势的大模型技术正在医疗领域崭露头角

得益于强劲算力与海量数据的加持，大模型技术近年来正以更强的学习性能和更优的模型拟合效果，在 AI 助力各行各业实施数字化转型的浪潮中凸显更为重要的作用。与传统 AI 模型相比，拥有巨大参数规模（动辄数百至上千亿参数量级）的大模型不仅具备更强的学习性能以及更优的模型拟合效果，其高效的迁移学习能力也能帮助用户实现在通用模型上完成不同类型的任务。同时，对思维链（Chain of Thought，CoT）的良好支持，也使大模型应用补齐了传统 AI 在逻辑推理能力上的短板。

上述优势，不仅推动 ChatGPT 等热门应用成为人们关注的焦点，也使大模型领域的技术与应用成为各大科技巨头争先探索的蓝海，推动其在社交、金融、电商以及医疗等垂直领域的迅速落地，并显露出可观的市场潜力。有相关预测数据表明，大模型市场在未来数年都将保持 21.4% 的年复合增长率（Compound Annual Growth Rate，CAGR），到 2029 年或达 408 亿美元的市场规模[73]。

一直以来，医疗行业都是 AI 技术落地的重要方向，而拥有更多优异特性的大模型技术显然可为智能医疗的推进带来更强动力。因此，在市场竞争激烈的医疗行业中，基于医疗大模型构建的各类 AI 应用也正逐渐在智慧医疗的各个应用场景中发挥作用，并获得医护人员、医患和管理者的认可。

无论是面向大众，提供普惠医疗服务的智能问答与家庭医疗助手，还是有助于医护人员提升效率的 AI 导诊和临床辅助诊疗应用，或是加速医疗影像处理效能，提高大病、恶疾早期发现率的 AI 阅片等，众多医疗 AI 企业正借助大模型，帮助医疗机构在诊疗服务全流程中实现更高效的服务能力、更精准的结果输出以及更广泛的运用范围。

在今天的医疗领域，在诸多企业、高校和科研机构的推动下，无论是医学科研、药物研发，还是智慧诊疗、医院管理，均有着基于大模型的应用涌现。而这其中，一些深耕医疗信息化多年，具有强大医疗 AI 应用研发能力和头部优势的医疗科技企业，同样也在这一趋势中将大模型作为其实现技术再突破、服务再提升的重要抓手。

常见的医疗大模型

根据应用领域和方向的不同，大模型通常会分为通用大模型、行业大模型、垂直大模型等不同类型。通用大模型通常不面向特定领域或任务，可在各种场景中都取得较好的应用。其也能作为基础模型，通过微调等方法来应对不同领域和任务的需求。而像医疗大模型这样的行业大模型，会更注重与医疗业务和医疗场景的结合，其学习和训练所用的数据通常都来自真实的医疗数据集，而其评估指标通常也会围绕医疗术语的识别、医疗知识问答的精准度来展开。

例如，除了常见的语义理解、逻辑推理、上下文记忆与理解以及数据安全等要素外，人们对医疗大模型的评估通常还要加入医学知识问答、导诊问诊、常见疾病风险评估、常见疾病辅助诊疗及依据以及病历生成等，力求覆盖更多常见医疗 AI 应用场景。为此，医疗大模型的训练数据通常会涵盖数以百万计的生物医学文献和数以亿计的健康记录和医疗数据。

目前医疗领域常见的基础大模型以及医疗大模型包括：

- **GPT**：作为一种生成式预训练 Transformer（Generative Pre-Trained Transformer）模型，是目前应用最流行的基础大模型；
- **ChatGLM-6B/ChatGLM2-6B**：由清华大学开源的一款支持中英双语问答的基础大模型，对中文处理实现了特别的优化，在医疗领域有着良好的运用；
- **Med-PaLM（谷歌医疗大模型）**：由谷歌与 DeepMind 推出的 Med-PaLM，在一些测评中被认为已与现实中人类临床医生的水平相当；
- **本草中文医学大模型**：由哈工大开源的医学智能问诊大模型，是基于 LLaMA-7B 模型的医学垂直领域模型，使用 CMeKG 中文医学知识图谱和 GPT3.5 API 构建；

[73] 数据援引自 Marketwatch 相关报告：https://www.marketwatch.com/press-release/large-language-model-llm-market-size-to-grow-usd-40-8-billion-by-2029-at-a-cagr-of-21-4-valuates-reports-7bbc5419

- DoctorGLM：基于 ChatGLM-6B 的中文问诊模型，通过中文医疗对话数据集进行微调，实现了包括 lora、p-tuningv2 等微调及部署。

以 ChatGLM-6B 大模型为例，该模型的结构如图 2-8-1 所示，其流水线回路主要包含 3 个主要模块，即 Embedding、GLMBlock 层和 lm_logits。模型的流水线中有两类不同的执行图，首次推理时不需要 KV 缓存作为 GLMBlock 层的输入。而从第二次迭代开始，QKV 注意力机制的上一次结果（pastKV）将成为当前一轮模型推理的输入。

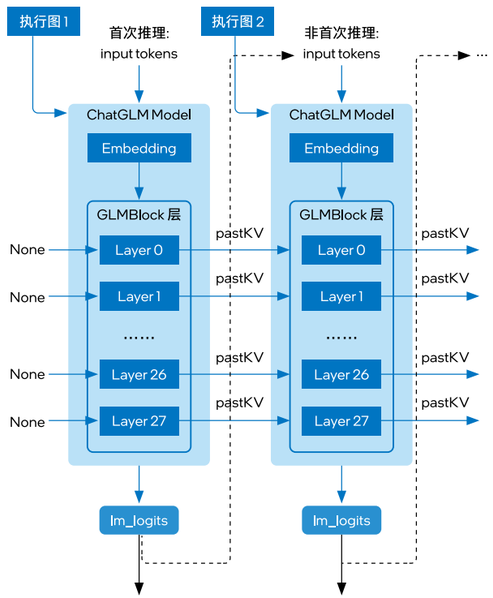


图 2-8-1 ChatGLM-6b 的模型结构

同时，医疗大模型通常还会借助大型医学知识库来提升专业性。例如本草中文医学大模型就引入了中文医学知识图谱 CMeKG（医学知识库围绕疾病、药物、检查指标等构建，字段包括并发症、高危因素、组织学检查、临床症状、药物治疗以及辅助治疗等）和 2023 年关于肝部恶性疾病的中文医学文献，并借助 GPT3.5 API 分别构造了大量问答数据和多轮对话的训练数据 [74]。

医疗大模型落地面临的挑战

但在医疗大模型的落地上，医疗机构和医疗科技企业还面临着一项挑战，即如何实现医疗大模型高质量、低成本的私有化部署。众所周知，医疗行业的特殊性使医疗机构对数据安全、隐私保护极为重视，任何医疗数据都不能离开安全可控的内网环境，所以医疗大模型需要进行私有化部署。同时，传统的大模型训练和推理工作通常借助专用芯片来完成，但其昂贵的价格不仅抬高了部署成本，同时普遍性的缺货或供货周期较长等问题也使方案的建设周期被大幅拉长，这些都阻碍着医疗大模型在医疗机构的普及。

数以百亿计的参数规模在给医疗大模型带来更优学习效果、更精准辅助诊疗结果的同时，也对承载平台的资源，包括算力、内存等提出了更严苛的要求。这一方面会影响 AI 应用的最终运行效率，降低医护、患者、管理者以及科研人员的使用体验，也会限制更大参数规模、更优性能的大模型在医疗机构的普及。

来自英特尔的软硬件产品与技术为医疗机构和医疗科技企业应对上述挑战提供了支持：

- 既包括推出新一代硬件产品在算力、内存性能上的巨大提升，用以满足医疗大模型在处理资源上的需求；
- 也包括借助软件和加速库来充分挖掘英特尔® 架构硬件平台潜能，并提升各个医疗大模型私有化部署时的运行速度。

英特尔产品与技术为大模型提供量化和非量化优化方案

基于英特尔产品与技术的量化优化方案

模型量化一直是提升大模型推理速度的重要优化方案之一。对 AI 模型的量化是指将训练好的模型的权值、激活值等从高精度数据格式（如 FP32 等）转化为低精度数据格式（如 INT4/INT8 等），这不仅可以降低推理过程中对内存等资源的需求，从而容纳更大参数规模的大模型，也能大幅提升推理速度，使医疗 AI 应用的运行更为迅捷。

在量化优化方案中，开发者可基于第四代英特尔® 至强® 可扩展处理器内置的指令集，借助由英特尔开发和开源的 BigDL-LLM 大模型库来实现推理加速量化方案。作为英特尔开源 AI 框架 BigDL 的一部分，BigDL-LLM 不仅提供了对各种低精度数据格式的支持和优化，也可基于不同处理器内置指令集（如英特尔® 高级矢量扩展 512_ 矢量神经网络指令（英特尔® AVX-512_VNNI）、英特尔® 高级矩阵扩展（英特尔® AMX）等）及其它软件实施推理加速，使大模型在英特尔® 架构平台上实现更高效的推理效能。

图 2-8-2 所示，在方案中，BigDL-LLM 可为医疗大模型提供了两种使用方法：便捷命令（Command Line Interface，CLI）方法和编程接口（Application Programming Interface，API）方法。在 CLI 命令模式下，开发者可方便地完成模型量化并评估量化后的推理效果，由此判断该量化方案是否适用于当前这个模型。这些 CLI 命令包括使用 llm-convert 命令来对模型的量化精度快速转换用于预览，或者使用 llm-cli/llm-chat 命令来运行并快速测试量化后的模型。

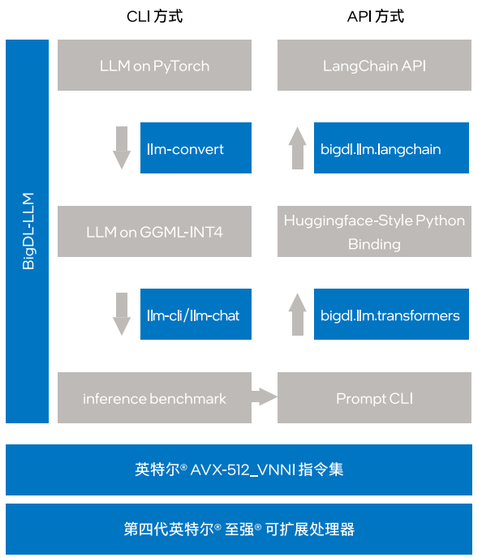


图 2-8-2 BigDL-LLM 为医疗大模型提供推理加速

另一方面，借助 BigDL-LLM 所提供的面向 HuggingFace 和 LangChain 的 API 编程接口，优化方案也能快速地将 LLM 量化方案整合进 HuggingFace 或 LangChain 的项目代码，进而便捷地完成模型部署。作为热门的 Transformers 开源库之一，HuggingFace 上的 Transformers 模型一直是各类大模型的重要组成部分，而导入 BigDL-LLM 的优势在于能让开发者改动少许代码（仅需更改 import，并在 from_pretrained 参数中设置 load_in_4bit=True 即可），在使用 bigdl.llm.transformers 后 BigDL-LLM 会在加载模型的过程中对模型进行 INT4 的低精度量化，由此实现对基于 HuggingFace Transformers 的模型进行加速。参考代码如下：

```
#基于 INT4 优化方案导入（load）Hugging Face Transformers 模型
from bigdl.llm.transformers import AutoModelForCausalLM
model = AutoModelForCausalLM.from_pretrained('/path/to/model/', load_in_4bit=True)

#在英特尔架构处理器上运行优化后的模型
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(model_path)
input_ids = tokenizer.encode(input_str, ...)
output_ids = model.generate(input_ids, ...)
output = tokenizer.batch_decode(output_ids)
```

与此同时，LangChain 也是近年来大模型领域流行的开源框架之一，BigDL-LLM 同样也通过 API 编程接口 bigdl.llm.langchain 提供了便于使用的 LangChain 集成能力，让开发者能轻松借助 BigDL-LLM 来从头开发或迁移基于 HuggingFace Transformers 优化的 INT4 模型或是其它原生 INT4 模型。以下是一个使用 LangChain API 运行 HuggingFace Transformers 模型（已进行 INT4 量化优化）的代码示例：

```
from bigdl.llm.langchain.llms import TransformersLLM
from bigdl.llm.langchain.embeddings import TransformersEmbeddings
from langchain.chains.question_answering import load_qa_chain

embeddings = TransformersEmbeddings.from_model_id(model_id=model_path)
bigdl_llm = TransformersLLM.from_model_id(model_id=model_path, ...)

doc_chain = load_qa_chain(bigdl_llm, ...)
output = doc = chain.run(...)
```

基于英特尔产品与技术的非量化优化方案

在量化优化方案之外，英特尔也通过 OpenVINO™ 工具套件提供了非量化优化方案。在帮助开发者使用 OpenVINO™ 工具套件的 Pipeline 构建医疗大模型的高效推理服务部署之余，

[74] 更多本草中文医学大模型，请参阅：https://github.com/scir-hi/huatuo-llama-med-chinese

英特尔还借助工具套件来助力优化模型推理流水线，通过消减模型输入和输出之间的内存副本来降低资源消耗，提升推理效率，并通过执行图的重新设计来优化模型中的组件。

例如在 ChatGLM-6B 这样的大模型运行过程中，随着所生成 tokens 长度不断增加，流水线推理过程中的模型输入和输出之间，将存留海量的大型内存副本（内存拷贝开销由模型的参数 hidden_size 以及迭代的次数决定），不仅将占据大量的内存空间，庞大的内存拷贝开销也会使推理的执行效率遭遇挑战。为此，基于 OpenVINO™ 工具套件的非量化优化方案执行了三个方面的优化动作。

首先是利用零拷贝（Zero-Copy）视图来传递预分配的 KV 所需的内存副本空间。由于传统的内存拷贝需要耗费大量的处理器资源和内存带宽，因此当内存副本规模大幅增加时，会成为大模型推理效率的瓶颈。而零拷贝技术的引入，能避免数据的多次拷贝，有效实现 KV 缓存加速。

第二项优化点包括使用 OpenVINO™ opset 来重构 ChatGLM 的模型架构，这能够帮助模型中的节点利用英特尔® AMX 指令集内联和多头注意力（Multi-Head Attention，MHA）融合来实现推理优化。如图 2-8-3 所示，优化方案构建的 OpenVINO™ stateful 模型在 GLMBlock 层重新封装了一个类，并按图中工作流来调用 OpenVINO™ opset，并通过其将图形数据序列化为中间表示（Intermediate Representation，IR）模型（如 .xml、.bin）。

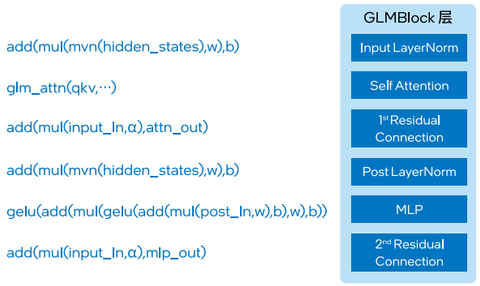


图 2-8-3 构建 OpenVINO™ stateful 模型

该优化方案一方面构建了全局的上下文结构体，用于在模型内部追加并保存每一轮迭代后的 pastKV 结果，减少相应的内存拷贝开销。另一方面，方案也通过内联优化（Intrinsic Optimization）的方式来实现了 Rotary Embedding 和 MHA 融合。

第四代英特尔® 至强® 可扩展处理器内置英特尔® AMX 指令集的引入，也能帮助 ChatGLM 等医疗大模型提升 BF16 或 INT8 精度数据格式下的模型推理速度。英特尔® AMX 指令集提供的内联指令能快速处理 BF16 或 INT8 精度数据格式的矩阵乘加运算，实现对 ChatGLM 模型中 Attention，Rotary Embedding 等算子的融合，在保证精度的同时提高运算效率，加速推理速度。

第三项优化是引入了 OpenVINO™ 工具套件在 HuggingFace 上的 Optimum 接口。Optimum 是 Huggingface Transformers 库提供的一个扩展包，可用来提升模型在特定硬件基础设施上的训练和推理性能。基于 OpenVINO™ 工具套件提供的 Optimum 接口，开发者能便捷地在提高性能之余，将模型扩展到更多医疗大模型推理应用中去。

这一优化同样也仅需改动少许代码即可完成，代码修改示例如下（面向文本分类（text-classification）任务）：

```
#替换 import 内容
#from transformers import AutoModelForSequenceClassificatdon
from optimum.intel import OVModelForSequenceClassificatdon
from transformers impoxt AutoTokenizer, pipeline

#替换模型内容
model_id ="distilbert-base-uncased-finetuned-sst-2-english"
#model = AutoModelForSequenceClassificatdon.from_pretrained(model_id)
model = OVModelForSequenceClassificatdon.frompretradned(model_id, export=True)
tokenizer = AutoTokenizer.from_pretrained(model_id)
cls_pipe = pipeline("text-classification", model=model, tokenizer=tokenizer)
outputs = cls_pipe("He's a dreadful magician.")
```

与此同时，该优化方法在其它任务，包括 token-classification、question-answering、audio-classification 以及 image-classification 等中也同样适用。

英特尔医疗大模型优化方案在惠每科技的实战

案例背景

随着基于医疗大模型的 AI 应用在医疗等领域赢得更广泛地实用化落地，更多医疗机构也正通过这些新型应用的引入来加速智慧医疗的进程。这其中，领先的医疗人工智能解决方案提供商北京惠每云科技有限公司（以下简称“惠每科技”）也以其临床决策支持系统（Clinical Decision Support System，CDSS）产品和海量医疗数据为基础，积极引入大模型技术来为医疗机构打造更高品质的医疗 AI 应用。

一直以来，惠每科技都以其领先的 CDSS 产品（如医院端核心应用 Dr.Mayson、临床科研平台 Darwin 等），通过实时数据分析与事中智能提示等核心能力的打造，助力医疗机构在临床诊疗决策、病案与病历管理、诊疗风险预警以及医保费用管理等环节中提升服务、诊疗和管理效能。

这些场景中对自然语言处理（Natural Language Processing，NLP）、计算机视觉（Computer Vision，CV）等 AI 能力的需求，无疑也给大模型技术提供了发挥所长的广阔空间。因此，惠每科技计划通过医疗大模型技术来构建更多富有价值的医疗 AI 应用，其与 CDSS 产品深度结合，助力数以百计的医疗机构用户进一步提升医疗服务质量。

案例描述与成效

如图 2-8-4 所示，在惠每科技最新发布的 CDSS 3.0 架构中，已将医疗大模型集成在新一代 AI 大数据处理平台中。这些医疗大模型不仅融合了惠每科技在医学知识库和专家系统上雄厚的知识积累，也聚集了其在 700 多家医疗机构落地应用的丰富实战经验，通过海量医疗数据在一系列大模型上重新训练而成，并已在病历生成等场景中成功运用。

为帮助医疗机构在保证服务质量和工作效率的情况下，以低成本方式实现医疗大模型的私有化落地，惠每科技与英特尔一起，在以第四代英特尔® 至强® 可扩展处理器为核心的硬件基础设施上，引入基于 BigDL-LLM 大模型开源库与 OpenVINO™ 工具套件的两种大模型推理加速方案，并在多个医疗机构开展部署。

首先，以第四代英特尔® 至强® 可扩展处理器为基础的量化加速方案，使用处理器内置的英特尔® AVX-512_VNNI 指令集，实现其医疗大模型在 INT4 低精度数据格式上的推理加速。其次，惠每科技也通过与英特尔的协作，通过 OpenVINO™ 工具套件实现了非量化的推理加速方案。目前，基于优化后的惠每医疗大模型所构建的医疗 AI 产品与应用，已在多家合作医疗机构进行了部署与运行，并取得了不错的效果。

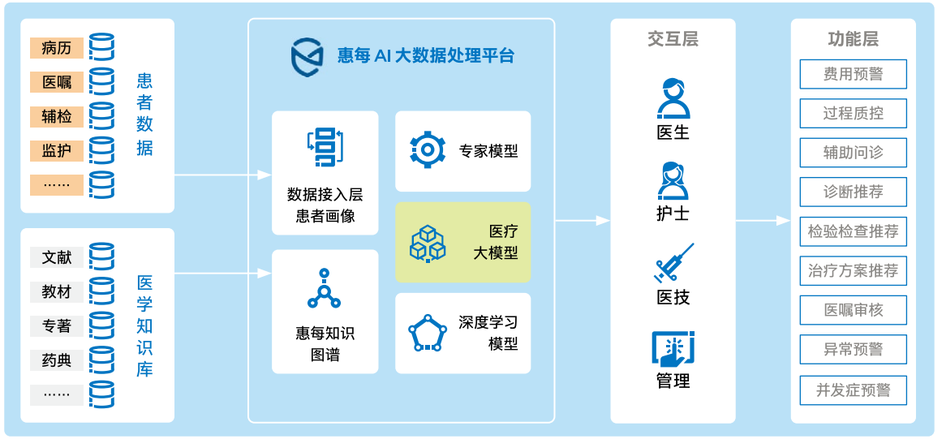


图 2-8-4 集成大模型理念的惠每新一代 AI 大数据处理平台架构

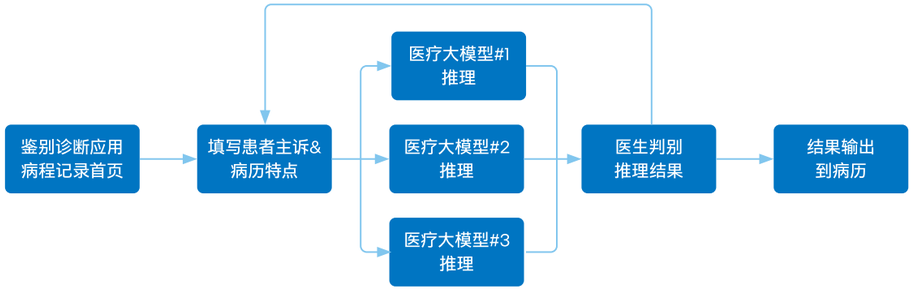


图 2-8-5 基于惠每医疗大模型构建的鉴别诊断应用工作流程

以基于大模型的鉴别诊断为例，这一辅助诊疗应用能体现临床医生的诊断思维链，而非简单的记录。如图 2-8-5 所示，医生在应用中打开病程记录首页并填写患者主诉及病历特点后，后台的 3 个不同医疗大模型就会迅速执行推理，在数秒后就生成鉴别诊断。医生可以点击查看不同大模型生成的结果，根据自身的专业意见选择最优结果，选择【一键回填】或复制粘贴到病历相应的位置。

在此过程中，医生可对病历生成的结果进行【点赞】/【点踩】，也可在系统中反馈错误或问题，或返回病程记录页继续修改患者主诉或病历特点，再次通过医疗大模型进行计算，进行新的鉴别诊断推理。这些设计能有效收集医生反馈，实现大模型的增强学习。

在合作医疗机构落地的另一项重要应用是出院记录的自动生成。传统上，诸如出院记录一类的流程，需要医院多个部门对多类数据进行总结并形成摘要，繁琐且容易出现差错。借助医疗大模型的技术优势，医生打开或保存【出院记录】时，会立即触发大模型后台计算。在数秒内得到结果后，医生即可查看包含出院诊断、入院情况、诊疗经过、出院情况和出院医嘱等内容的结果。医生可【一键回填】或复制粘贴到病历相应的位置，可对病历生成的结果进行【点赞】/【点踩】，也可点【识别错误】反馈相应问题。

通过惠每科技与英特尔的协同优化，由惠每医疗大模型构建的医疗 AI 应用无论是在应用效率还是在准确性等方面都获得了提升，并很快在临床应用价值方面表现出了显著的优势。这些优势包括：

- **医疗辅助诊疗准确性提升**：通过对大量医疗数据的有效学习，医疗大模型能持续学习各种疾病特征，并借助优化方案更快、更精准地做出判断。结合惠每科技医疗知识库，为医护人员提供更加科学和准确的辅助诊疗方案和建议，优化医疗决策；
- **医护与管理人员效率提升**：借助基于医疗大模型构建的各类医疗 AI 应用，医护人员可以高效地获取患者的辅助诊疗结果和病情 / 病历分析，从而能将更多时间和精力专注于患者的治疗和康复。同时医疗机构管理人员也能在诊疗风险预警、医保费用管理等环节上实现更为直观高效的管控。

上述方案中基于医疗大模型的应用，都能与惠每科技 CDSS 系统实现无缝衔接，并使用既有的英特尔® 架构处理器平台即可完成部署，而无需购置专用的加速芯片或加速服务器，从而有效降低大模型部署的成本压力。来自惠每科技的数据统计表明，在某合作医院一科室上线 1 个月后，鉴别诊断应用的使用率已达 23% 以上，出院记录自动生成应用的使用率达到 15% 以上，说明基于医疗大模型构建的 AI 能力已获得医生的初步认可。[75]

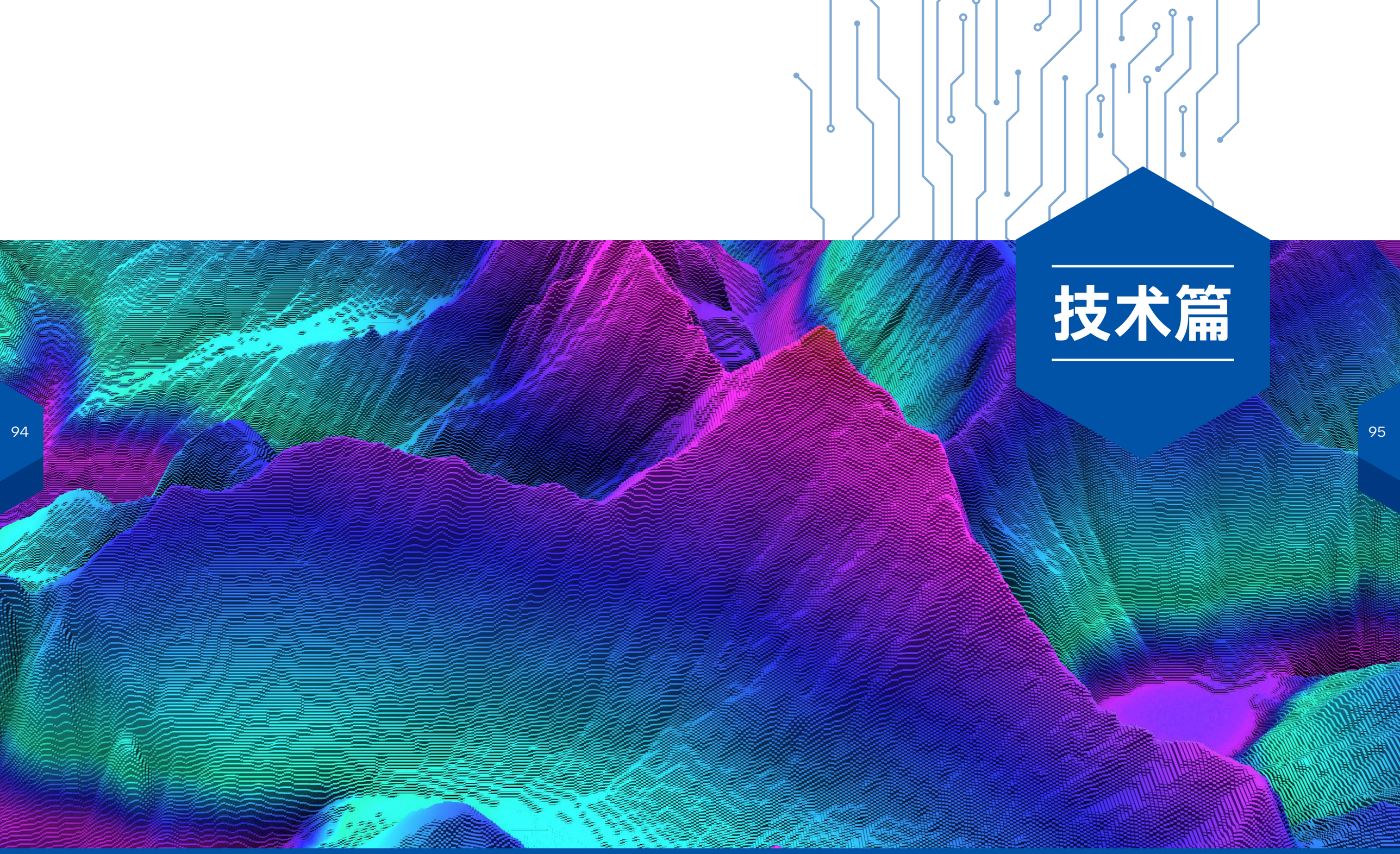
为评估优化后的医疗大模型效果，惠每科技参加了由中国健康信息处理大会（China Health Information Processing Conference, CHIP）组织的中文临床医疗信息处理权威评测。这一评测全部使用中文真实医疗数据，覆盖多个常见医疗 AI 应用场景，如医疗术语识别、医疗知识问答等，并采用量化的 F1 值进行排名。同时在大模型评测中，必须使用一个大模型同时完成 16 个任务的考验，非常具有挑战性。最终惠每科技从 396 支参赛队伍中脱颖而出，荣获 “CHIP2023-PromptCBLUE 医疗大模型评测” 参数高效微调赛道第一名。[76]

[75] 数据引自惠每科技数字医学云讲坛第 141 期，详细信息请访问：
https://www.e-chinc.com/#/ResourcesDetailVideo?id=1704682818727731202&packId=1614869950189219841
[76] 数据引自天池官网：https://tianchi.aliyun.com/competition/entrance/532132/rankingList。

小结

大模型等 AI 技术，正在像水、电等基础设施能力一样，将在医疗机构的未来医疗服务体系中产生无可替代的价值。这一进程中，更多医疗机构也正借助 AI、大数据等新技术、新能力来加速智慧医疗的进程，提升诊疗效率，提高服务质量。为帮助医疗机构和医疗科技企业更好地挖掘医疗大模型的潜能，令基于医疗大模型构建的各类应用更具效率，英特尔也通过第四代英特尔® 至强® 可扩展处理器、BigDL-LLM 大模型开源库和 OpenVINO™ 工具套件等软硬件产品与技术，推动使医疗大模型的推理效能在广泛应用的 IT 平台上获得提升，从而让医疗机构能以高质量、低成本的方式获得大模型的私有化部署。

面向未来，英特尔还将与更多合作伙伴一起，积极探索大模型技术在医疗领域中更广泛、更深入的应用，例如利用大模型开展病历内涵质控等，进而推动医疗全流程的 AI 技术或智能化加持，让智慧医疗惠及更多医与患，从而普惠大众。



技术篇

第四代英特尔® 至强® 可扩展处理器

第四代英特尔® 至强® 可扩展处理器旨在为人工智能、数据分析、存储和科学计算方面快速增长的工作负载提供性能加速。该处理器具备多种内置加速器，帮助客户将零信任安全策略付诸实践，同时利用先进的安全技术，即使面对敏感或受监管的数据，也能解锁新的商业合作机会和洞察。使用这款处理器可跨多个云和边缘环境进行扩展，满足自身的部署需求。英特尔® 至强® 可扩展处理器具有很强的灵活性，可在其上选择不同的云服务，帮助企业顺利实现应用移植。

基础性能进一步大幅提升

- 第四代英特尔® 至强® 可扩展处理器采用全新架构，单核性能比上一代产品更高，每路配备多达 60 个内核。每个系统支持单路、双路、四路或八路配置。为了与内核数增加这种情况相匹配，该平台在内存和 I/O 子系统方面也做了相应改进。DDR5 内存提供的带宽和速度与 DDR4 相比提高多达 1.5 倍，速率达到 4,800 MT/s[1]。此外，该平台还具有每路 80 条 PCIe Gen5 通道的特点，与之前的平台相比，I/O 得到显著提升。本代处理器还可提供 CXL 1.1 连接，支持高网络带宽并使附加加速器能够高效运行。第四代英特尔® 至强® 可扩展处理器支持的技术支持根据工作负载要求的变化灵活扩展和调整。此外，本代处理器还可助力实现以下优势：
- 进一步提升网络、存储和计算性能，并通过将繁重的任务卸载到英特尔® 基础设施处理单元（Intel® Infrastructure Processing Unit，英特尔® IPU）来提高 CPU 利用率；
- 通过英特尔® UPI 2.0 提高多路带宽（高达 16 GT/s）；
- 使用英特尔® Speed Select 技术（英特尔® SST）调整 CPU 配置，满足特定工作负载的需求；
- 增加三级缓存（LLC）共享容量（所有内核共享多达 100 MB LLC）；
- 通过硬件增强型安全功能加强对安全态势的掌控；
- 使用英特尔® Virtual RAID on CPU（英特尔® VROC），从而无需再用单独的 RAID 卡。

第四代英特尔® 至强® 可扩展处理器的新特性或新功能

PCI Express Gen5（PCIe 5.0）

带来全新的 I/O 速度，可在 CPU 和互联设备之间实现更高的吞吐量。第四代英特尔® 至强® 可扩展处理器具有多达 80 条 PCIe 5.0 通道，非常适合高速网络、高带宽加速器和高性能存储设备。PCIe 5.0 的 I/O 带宽是 PCIe 4.0 的两倍，仍具备向后兼容性并提供用于 CXL 连接的基础插槽[2]。

DDR5

以更高内存带宽克服数据瓶颈，提高计算性能。与 DDR4 相比，DDR5 的带宽提高多达 1.5 倍，因此有机会提升性能、容量和能效并降低成本[3]。借助 DDR5，第四代英特尔® 至强® 可扩展处理器提供的速率可高达 4,800 MT/s（1DPC）或 4,400 MT/s（2DPC）。

CXL

借助面向下一代工作负载的 CXL 1.1，降低数据中心的计算时延并帮助减少 TCO。CXL 是另一种跨标准 PCIe 物理层运行的协议，可以在同一链路上同时支持标准 PCIe 设备和 CXL 设备。CXL 可带来的一大关键能力是在 CPU 和加速器之间创建统一且一致的内存空间，它将革新未来数年数据中心服务器架构的构建方式。

内置众多加速引擎，重新定义性能

与增加 CPU 内核数相比，内置加速器是一种提升性能更有效的方法。其不但可以提高 CPU 利用率，降低功耗，并提高投资回报率（ROI），同时还能帮助企业实现可持续发展目标。英特尔® 至强® 可扩展处理器支持广泛且独特的内置加速器，有助于提高性能和效率，减少另行添置专用硬件的需求。在云端和本地环境中，这些专用功能支持人工智能、安全性、科学计算、数据分析、存储和网络等目前最为常见的严苛工作负载。

- **英特尔® 高级矩阵扩展（Intel® Advanced Matrix Extensions，英特尔® AMX）**可加速自然语言处理（NLP）、推荐系统和图像识别等深度学习（DL）推理和训练工作负载。
- **英特尔® 数据流加速器（Intel® Data Streaming Accelerator，英特尔® DSA）**可通过优化流数据的传输和转换操作，大幅提升存储、网络和数据密集型工作负载的性能。
- **英特尔® 存内分析加速器（Intel® In-Memory Analytics Accelerator，英特尔® IAA）**可提高数据分析性能，同时从 CPU 内核上卸载任务，为数据库查询及其他工作负载加速。
- **英特尔® 动态负载均衡器（Intel® Dynamic Load Balancer，英特尔® DLB）**可随系统负载的变化将网络数据动态地分配到多个 CPU 内核上，基于硬件高效实现负载平衡。
- **面向 vRAN 的英特尔® 高级矢量扩展（Intel® Advanced Vector Extensions，英特尔® AVX）**在相同功耗范围内可将虚拟无线接入网络（vRAN）的密度较上一代产品提高多达 2 倍[4]。
- **英特尔® 高级矢量扩展 512（Intel® Advanced Vector Extensions 512，英特尔® AVX-512）**支持多达两个融合乘加（FMA）单元，并包含多项优化，可为要求严苛的计算任务提高性能。
- **英特尔® 数据保护与压缩加速技术（英特尔® QAT）**可加速加解密和数据压缩，它通过从处理器内核卸载这些任务，帮助降低系统资源消耗。
- **英特尔® 密码操作硬件加速（Intel® Crypto Acceleration）**降低了实施普遍数据加密的影响，并提高了安全套接字层（SSL）Web 服务器、5G 基础设施和 VPN/防火墙等加密敏感型工作负载的性能。

英特尔® 至强® 可扩展处理器 PLATINUM	英特尔® 至强® 可扩展处理器 GOLD	英特尔® 至强® 可扩展处理器 SILVER
多达 8 路的可扩展性	多达 4 路的可扩展性	多达 2 路的可扩展性
4 个英特尔® UPI 端口，速率为 16 GT/s	3 个英特尔® UPI 端口，速率为 16 GT/s	2 个英特尔® UPI 端口，速率为 16 GT/s
80 条 PCIe 5.0 通道 + CXL	80 条 PCIe 5.0 通道 + CXL	80 条 PCIe 5.0 通道 + CXL
支持 DDR5，速率高达 4,800 MT/s（每通道 1 个 DIMM）或 4,400 MT/s（每通道 2 个 DIMM）	支持 DDR5，速率高达 4,800 MT/s（每通道 1 个 DIMM）或 4,400 MT/s（每通道 2 个 DIMM）	支持 DDR5，速率高达 4,800 MT/s（每通道 1 个 DIMM）或 4,400 MT/s（每通道 2 个 DIMM）
支持英特尔® 傲腾™ 持久内存 300 系列	支持英特尔® 傲腾™ 持久内存 300 系列	英特尔® AVX-512（两个 512 位 FMA）
英特尔® AVX-512（两个 512 位 FMA）	英特尔® AVX-512（两个 512 位 FMA）	英特尔® 超线程技术和英特尔® 睿频加速技术
英特尔® 超线程技术和英特尔® 睿频加速技术	英特尔® 超线程技术和英特尔® 睿频加速技术	英特尔® 深度学习加速技术和英特尔® AMX
英特尔® AMX	英特尔® 深度学习加速技术和英特尔® AMX	英特尔® SGX 最大飞地容量高达 64 GB
英特尔® SST	英特尔® SST	可通过英特尔® QAT、英特尔® DLB、英特尔® DSA 和英特尔® IAA 加速工作负载
先进的可靠性、可用性和可维护性（RAS）	先进的 RAS	
英特尔® SGX 最大飞地容量高达 128 GB（在特定型号的 SKU 上最大飞地容量高达 512 GB）	英特尔® SGX 最大飞地容量高达 128 GB	
可通过英特尔® QAT、英特尔® DLB、英特尔® DSA 和英特尔® IAA 加速工作负载	可通过英特尔® QAT、英特尔® DLB、英特尔® DSA 和英特尔® IAA 加速工作负载	

[1]、[2]、[3] https://www.intel.cn/content/www/cn/zh/products/docs/processors/xeon-accelerated/4th-gen-xeon-scalable-processors-product-brief.html

[4] https://www.intel.cn/content/www/cn/zh/products/docs/processors/xeon-accelerated/4th-gen-xeon-scalable-processors-product-brief.html

英特尔® 至强® 铂金 8400 处理器是打造安全且敏捷的混合云数据中心的基石，专为高级数据分析、AI、高密度基础设施和多云工作负载而设计。这些处理器具备更高性能、强大的平台功能和出色的工作负载加速能力。它们具有更出色的基于硬件的安全性和强大的多路处理性能——特定型号的英特尔® 至强® 铂金 8400 处理器支持多达 8 路配置。借助值得信赖且经过硬件增强的数据服务交付以及全新的 I/O 和连接技术，这些处理器在 I/O、内存、存储和网络技术方面均实现提升，因而能够更好地在数据驱动程度日益加深的世界中挖掘可执行洞察。相关提升包括：

- 每个英特尔® 至强® 可扩展处理器具备多达 60 个内核；
- 每个处理器有 8 条内存通道，速率高达 4,800 MT/s (1 DPC)；
- 英特尔® AMX 支持下的 AI 加速带来深度学习推理和训练性能的巨大飞跃。

AI：凭借更优的矢量指令和矩阵乘法运算，第四代英特尔® 至强® 可扩展处理器展现出更为出色的 AI 推理和训练性能。英特尔® AMX 可以显著提高推荐系统、NLP、图像识别、媒体处理和分发以及媒体分析等深度学习工作负载的性能。

- AI 性能数据请参考：英特尔® AI 数据中心产品的性能数据

Framework Version ☑ Model ☑ Usage ☑ Precision ☑ Throughput ☑ Perf/Watt ☑ Accuracy ☑ Latency(ms) ☑ Batch size ☑ Config*

Framework Version	Model	Usage	Precision	Throughput	Perf/Watt	Accuracy	Latency(ms)	Batch size	Config*
Intel PyTorch 2.0 (master)	GPT-J 6B LAMBADA (32 tokens input)	LLM / chatGPT	INT8	19.2 tokens/s		77.89(%)	51	1	1 instance per socket, 1 socket
Intel PyTorch 2.0 (master)	GPT-J 6B LAMBADA (32 token input)	LLM / chatGPT	BF16	14.5 tokens/s			68	1	1 instance per socket, 1 socket
Intel PyTorch 2.0 (master)	GPT-J 6B LAMBADA (32 tokens input)	LLM / chatGPT	BF16	84.5 tokens/s			210	16	1 instance per socket, 1 socket
Intel PyTorch 2.0 (master)	T5-3B Samsum (32 tokens input)	LLM / chatGPT	BF16	35.9 tokens/s			33	1	1 instance per socket, 1 socket
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	Int8				0.89	1	1 instance per socket, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	int8	9680.59 img/s	9.46		2.89	1	4 cores/instance, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	Int8	13931.2 img/s				64	14 cores/instance, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	int8	13012.99 img/s	12.78	75.99(%)		116	1 instance per socket, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	bf16				1.25	1	56 cores/instance, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	Bf16	7362.99 img/s				32	14 cores/instance, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	bf16	7002.92 img/s	6.81	76.14(%)		68	1 instance per socket, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	bf32	2068.72 img/s		76.13 (%)		64	1 instance per socket, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	fp32	1444.76 img/s				256	2 cores/instance, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	fp32				2.7	1	1 instance per socket, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	fp32	1293.33 img/s	1.25		21.65	1	4 cores/instance, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	fp32	1338.80 img/s	1.25	76.13(%)		64	1 instance per socket, 2 sockets
Intel PyTorch 1.13	ResNet50 v1.5	Image Recognition	bf16	5805.89 img/s	5.63		4.82	1	4 cores/instance, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	int8	12390.26 img/s	11.74	76.02(%)		116	1 instance per socket, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	int8	8221.7 img/s	8.52		3.41	1	4 cores/instance, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	bf16	6299.25 img/s	5.99	76.75(%)		80	1 instance per socket, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	bf16	5451.4 img/s	5.43		5.14	1	4 cores/instance, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	bf32	1984.48 img/s		76.47(%)		64	1 instance per socket, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	fp32	1238.55 img/s	1.2		22.61	1	4 cores/instance, 2 sockets
Intel TensorFlow 2.11	ResNet50 v1.5	Image Recognition	fp32	1294.17 img/s	1.25	76.48(%)		64	1 instance per socket, 2 sockets
OpenVINO	ResNet50 v1.5	Image Recognition	int8	11951.34 img/s	11.6			64	1 instance per socket, 2 sockets
OpenVINO	ResNet50 v1.5	Image Recognition	int8	8982.73 img/s	8.92	76.36(%)		1	4 cores/instance, 2 sockets
OpenVINO	ResNet50 v1.5	Image Recognition	bf16	5719.37 img/s	5.66	76.47(%)		1	4 cores/instance, 2 sockets
OpenVINO	ResNet50 v1.5	Image Recognition	bf16	6087.33 img/s	5.93			116	1 instance per socket, 2 sockets
OpenVINO	ResNet50 v1.5	Image Recognition	fp32	1252.31 img/s	1.23	76.46(%)		1	4 cores/instance, 2 sockets
OpenVINO	ResNet50 v1.5	Image Recognition	fp32	1248.72 img/s	1.21			64	1 instance per socket, 2 sockets

访问链接了解更多第四代英特尔® 至强® 可扩展处理器详情
https://www.intel.cn/content/www/cn/zh/products/docs/processors/xeon-accelerated/4th-gen-xeon-scalable-processors.html

扫码了解更多第四代英特尔® 至强® 可扩展处理器详情

扫码了解第四代英特尔® 至强® 可扩展处理器的配置和 AI 调优指南

第三代英特尔® 至强® 可扩展处理器

英特尔对面向四路和八路的第三代英特尔® 至强® 可扩展处理器（Cooper Lake）和面向单路和双路的第三代英特尔® 至强® 可扩展处理器（Ice Lake）在多样化的工作负载类型和性能需求方面进行了优化。

基础性能

- 第三代英特尔® 至强® 可扩展处理器基于平衡、高效的架构构建，该架构可提升内核性能、内存和 I/O 带宽，为处理从数据中心到边缘的各种工作负载提速。在单路和双路配置中，支持每处理器多达 40 个内核，在四路和八路配置中则支持每处理器达 28 个内核，在八路配置下，单平台支持多达 224 个内核；
- 单个处理器支持 8 条 DDR4 内存通道（Ice Lake）或 6 条 DDR4 内存通道(Cooper Lake)，最高速率为 3,200 MT/s。同时每路多达 64 条 PCI Express Gen4 通道，实现更高的每核 I/O 带宽；
- 多达 6 条英特尔® 超级通道互联（英特尔® UPI）通道有效提高了平台可扩展性以及 I/O 密集型工作负载的 CPU 间带宽，从而在提高吞吐量和能效之间达成平衡。

增强的 AI 加速与安全能力

- 第三代英特尔® 至强® 可扩展处理器加入了基于英特尔® AVX - 512 的增强版英特尔® 深度学习加速技术，同时支持 16 位 Brain Floating Point (BF16) 和矢量神经网络指令 (VNNI)，有效加速人工智能推理和训练性能。其中 BF16 适用于特定型号的第三代英特尔® 至强® 可扩展处理器，其在视觉、自然语言处理和强化学习等需要兼顾吞吐量和准确率的 AI 应用场景可以提供更有效的训练与推理加速能力。而矢量神经网络指令 (VNNI) 能够充分提高计算资源和缓存的利用率、减少潜在的带宽瓶颈，以此增强推理工作负载；
- 单路和双路配置的第三代英特尔® 至强® 可扩展处理器对英特尔® SGX 提供支持，帮助用户无论是从边缘到数据中心还是到多租户公有云，都可以在更好地保护数据和应用代码安全的前提下，采用联邦学习等方法，以多源数据加强 AI 应用的应用效能。

自定义性能助推各种工作负载

- 第三代英特尔® 至强® 可扩展处理器增强了英特尔® SST（英特尔® Speed Select 技术）功能，其可以对处理器性能实施精细控制，有助于优化 TCO。大部分第三代英特尔® 至强® 铂金和金牌处理器都支持英特尔® SST– BF、英特尔® SST –CP 和英特尔® SST – TF 等不同模式的 SST，而第三代英特尔® 至强® 可扩展处理器 Y SKU 支持新的英特尔® SST- PP 模式，可以为用户提供更多内核、频率、外形尺寸和功率配置选择。

适用于不同工作负载的第三代英特尔® 至强® 可扩展处理器

- 英特尔® 至强® 铂金 8300 处理器是打造可靠、敏捷的混合云数据中心的基石。处理器具备增强型硬件安全功能以及出色的多路处理性能，适用于关键业务的实时分析、机器学习、人工智能、科学计算和多云工作负载。
- 英特尔® 至强® 金牌 6300 和 5300 处理器支持更高的内存速度、更大的内存容量以及多达四路的可扩展性，带来更出色的性能和内存功能、硬件增强型安全和工作负载加速。
- 英特尔® 至强® 银牌 4300 处理器提供基本性能、更快的内存速度以及更高的能效，为入门级数据中心计算、网络和存储带来所需的硬件增强性能。

访问链接了解更多第三代英特尔® 至强® 可扩展处理器详情
https://www.intel.cn/content/www/cn/zh/products/docs/processors/xeon/3rd-gen-xeon-scalable-processors-brief.htmlhttps://www.intel.cn/content/www/cn/zh/products/docs/processors/xeon/3rd-gen-xeon-scalable-processors-brief.html

扫码了解更多第三代英特尔® 至强® 可扩展处理器详情

英特尔® 至强® CPU Max 系列

过去十年，随着人工智能技术的加入，峰值算力大幅增长，但由于在向内核传输数据时效率低，因此工作负载性能未能同步提升。英特尔® 至强® CPU Max 系列的诞生，使英特尔® 至强® 平台如虎添翼，它是英特尔唯一一个基于 x86 架构并采用高带宽内存（HBM）的 CPU 系列，可释放和加速内存密集型科学计算和 AI 工作负载。

更高带宽，更优性能

英特尔® 至强® CPU Max 系列采用全新微架构，支持一系列可提升平台能力的特性，包括更多内核、先进的 I/O 与内存子系统，以及可加速重大发现的内置加速器。英特尔® 至强® CPU Max 系列具有以下特性：

- **多达 56 个 P-core（性能核）：**内核由 4 个小芯片构成，采用英特尔的嵌入式多芯片互连桥接（EMIB）技术连接，功耗为 350 W；
- **64 GB 高带宽封装内存及 PCIe 5.0 和 CXL 1.1 I/O。**英特尔® 至强® CPU Max 系列每核均具备 HBM 容量，可满足大多数常见科学计算工作负载的要求；
- 与其他 CPU 相比，在使用 Numenta 的 AI 技术进行自然语言处理时，其 HBM 优势可带来高达 20 倍的性能提升 [5]。

加速科学创新

英特尔® 至强® CPU Max 系列能够与英特尔® 至强® 平台实现轻松整合，不但可以获得处理要求严苛的工作负载所需的性能与能效，还可得到各种出色的内置加速器（包括英特尔® AMX，英特尔® DSA 等，具体详见第 * 页详细介绍）的助力。利用面向科学计算和 AI 工作负载的关键加速器，提高 CPU 使用效率、降低功耗、实现更高的投资回报率（ROI）。另外，由于处理器插槽（Socket）配置相同，可轻松将英特尔® 至强® CPU Max 系列处理器添加到第四代英特尔® 至强® 可扩展平台，并且在大多数部署方案中都无需更改代码。

灵活应对各种科学计算和 AI 工作负载

英特尔® 至强® CPU Max 系列处理器具备出色的灵活性，可根据工作负载的特性，在不同的内存模式或配置下运行：

- **“仅 HBM”模式：**该模式支持内存容量需求不超过 64 GB 的工作负载以及每核 1 至 2 GB 的内存扩展能力，同时无需更改代码和另购 DDR，即可启动系统；
- **“HBM Flat”模式：**该模式可为需要大内存容量的应用提供灵活性，它通过 HBM 和 DRAM 提供一个平面内存区域（flat memory region），适用于每核内存需求大于 2 GB 的工作负载。使用该模式时可能需要更改代码；
- **“HBM 缓存”模式：**旨在提升内存容量需求大于 64 GB 或每核内存需求大于 2 GB 的工作负载的性能。使用该模式时，无需更改代码，且 HBM 可缓存来自 DDR 的事务。

英特尔® 至强® CPU Max 系列	
内核数	32-56
HBM2e 内存	64 GB
HBM 最大传输速率	3200 MT/s
DDR5最大传输速率	4800MT/s（1 个 DPC） 4400 MTs（2 个 DPC）
加速器	AMX，4 个英特尔® DSA
AI/ML指令	INT8 和 BFLOAT16

跨多架构加速科学计算和 AI 工作负载

整个英特尔® 至强® CPU Max 系列的产品均得到 oneAPI 的支持。oneAPI 是一个统一的、基于标准的开放式通用编程模型，可释放生产力并解锁性能。开发人员可利用英特尔® oneAPI 工具套件以及面向特定领域的专用工具套件，打造跨多种架构运行的通用计算、科学计算和 AI 应用，并对其进行分析、优化和扩展。这些资源包括矢量化、多线程、多节点并行和内存优化方面的前沿技术，可轻松构建随时能为科学计算所用的高性能、多架构软件。

扫码了解更多英特尔® 至强® CPU Max 系列详情

扫码了解英特尔® 至强® CPU Max 系列配置和调优指南

[5] https://www.intel.cn/content/www/cn/zh/products/docs/processors/xeon/xeon-max-series-product-brief.html

英特尔® 高级矢量扩展 512（英特尔® AVX-512）

快速分析日益增多的数据，并将其转化为有价值的洞察力，这种能力将为商业、科学研究乃至人们的日常生活创造新的机遇。英特尔® 至强® 可扩展处理器和英特尔® 至强融核™ 处理器产品家族，增添了旨在加速数据分析的创新功能。

当前的工作负载，通常需要在多个数据元素上执行同样的操作，在传统的“标量处理”时代，指令在同一时间，只能在一个单一数据元素上执行，以致在处理海量数据时极为耗时。认识到标量处理的不足之后，从上世纪 90 年代后期开始，英特尔开始将单指令多数据流（Single Instruction Multiple Data，SIMD）矢量功能整合到英特尔® 处理器中。英特尔® SSE 技术刚推出时，提供了 128 位寄存器和 SIMD 指令，可同时处理多达 4 个 32 位数据元素，大大加快了相关操作的处理速度。在此之后，英特尔® AVX 指令集和英特尔® AVX2 指令集又将寄存器宽度扩展了一倍，使相关操作的处理性能实现近乎翻倍的提升。

如今，英特尔® AVX – 512 指令集将矢量计算性能提升至新高度，寄存器的宽度和数量又在英特尔® AVX 指令集和英特尔® AVX2 指令集的基础上扩展了一倍，寄存器已由最初的 64 位升级到了 512 位，且具备两个 512 位的 FMA 单元，这意味着应用程序可同时执行 32 次双精度、64 次单精度浮点运算，或操作八个 64 位和十六个 32 位整数。

英特尔® 至强® 可扩展处理器可支持多种工作负载。英特尔® AVX-512 指令集通过矢量化性能提升，使更大数据集上的运算速度更快，满足包括科学计算在内的严苛计算任务的性能提升。例如在 OpenFOAM 在运行时，每个内核同时使用两个矢量处理单元（其中每个单元能同时处理 16 个单精度（32 位）或 8 个双精度（64 位）的浮点数）。

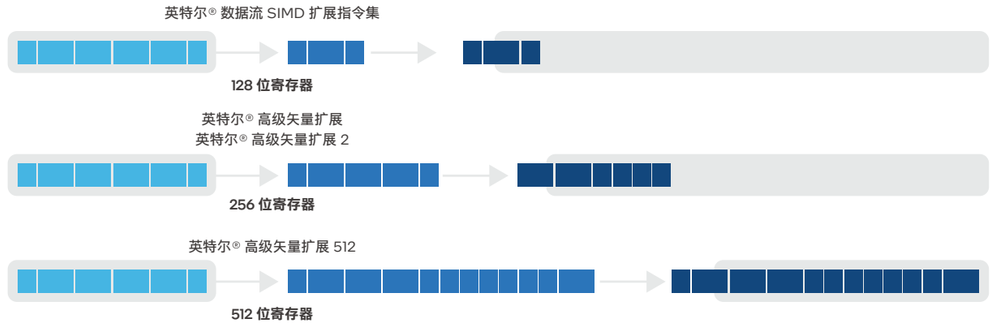


图 3-1 英特尔® SSE、英特尔 AVX2 和英特尔® AVX-512 之间的寄存器大小和计算效率的差异说明

英特尔® 高级矩阵扩展（英特尔® AMX）

第四代英特尔® 至强® 可扩展处理器内置 AI 加速器——英特尔® AMX，是企业和机构优化 AI 流水线的理想选择。平衡推理是 CPU 在 AI 应用中的主要用例，英特尔® AMX 专为该用例设计并且具备更多训练能力。目前，在所有运行 AI 推理工作负载的已装机数据中心处理单元中，英特尔® 至强® 可扩展处理器的占比高达 70%[6]；因此，为新的 AI 部署选择内置英特尔® AMX 的第四代英特尔® 至强® 可扩展处理器，是一种既高效又具有成本效益的 AI 工作负载加速方式。

英特尔® AMX 是什么？

英特尔® AMX 是内置于第四代英特尔® 至强® 可扩展处理器中的加速器，可优化深度学习（DL）训练和推理工作负载。借助英特尔® AMX，第四代英特尔® 至强® 可扩展处理器可在优化通用计算和 AI 工作负载间快速转换。开发人员可以编写非 AI 功能代码来利用处理器的指令集架构（ISA），也可编写 AI 功能代码，以充分发挥英特尔® AMX 指令集的优势。英特尔已将其 oneAPI DL 引擎——英特尔® oneAPI 深度神经网络库（Intel® oneAPI Deep Neural Network Library，英特尔® oneDNN）集成至包括 TensorFlow、PyTorch、PaddlePaddle 和 ONNX 在内的多个主流 AI 应用开源工具当中。

英特尔® AMX 架构

英特尔® AMX 架构由两部分组件构成：

- 第一部分为 TILE，由 8 个 1 KB 大小的 2D 寄存器组成，可存储大数据块；
- 第二部分为平铺矩阵乘法（TMUL），它是与 TILE 连接的加速引擎，可执行用于 AI 的矩阵乘法计算。

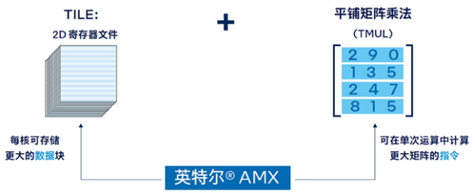


图 3-2 英特尔® AMX 架构由 2D 寄存器文件（TILE）和 TMUL 组成

英特尔® AMX 支持两种数据类型：INT8 和 BF16，两者均可用于 AI 工作负载所需的矩阵乘法运算。

- 当推理无需 FP32（AI 经常使用的单精度浮点格式）的精度时可使用 INT8 这种数据类型。由于该数据类型的精度较低，因此单位计算周期内运算次数就更多；
- BF16 这种数据类型实现的准确度足以达到大多数训练的要求，必要时它也能让 AI 推理实现更高的准确度。

凭借这种新的平铺架构，英特尔® AMX 实现了大幅代际性能提升。与运行英特尔® AVX-512_VNNI 的第三代英特尔® 至强® 可扩展处理器相比，运行英特尔® AMX 的第四代英特尔 ® 至强® 可扩展处理器将单位计算周期内执行 INT8 运算的次数从 256 次提高至 2,048 次。此外，如图 3-3 所示，第四代英特尔® 至强® 可扩展处理器可在单位计算周期内执行 1,024 次 BF16 运算，而第三代英特尔® 至强® 可扩展处理器执行 FP32 运算的次数仅为 64 次[7]。

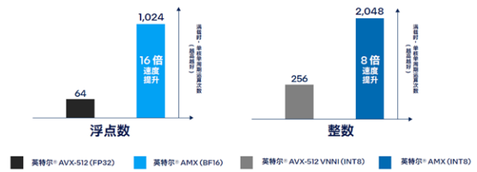


图 3-3 与英特尔® AVX - 512 _VNNI 相比，英特尔® AMX 在处理 INT8 和 BF16 两种数据类型时表现更出色[8]

使用英特尔® AMX 立启新加速

借助英特尔® AMX，几乎无需费力，即可提升性能。这得益于多个默认使用的框架都经过英特尔® oneDNN 的优化。Windows 和 Linux 操作系统、基于内核的虚拟机（KVM）和多个主流虚拟机管理程序都支持英特尔® AMX 指令集。INT8 和 BF16 运算在 TensorFlow 和 PyTorch 等开源框架内可自动优化。开发人员可借助英特尔® 分发版 OpenVINO™ 工具包（Intel® Distribution of OpenVINO™ Toolkit）实现 AI 推理的自动化、优化、微调和运行，且几乎或者完全不需要具备编码知识。而且，开发人员只需使用英特尔® Neural Compressor 便可将训练模型量化为 INT8 数据类型。

访问链接了解更多英特尔® AMX 的详情
https://www.intel.cn/content/www/cn/zh/customer-spotlight/cases/accelerate-ai-workload-with-amx.html

扫码了解更多英特尔® AMX 详情

[6] 基于英特尔对截至 2021 年 12 月运行 AI 推理工作负载的全球数据中心服务器装机容量的市场建模。
[7]、[8] https://www.intel.cn/content/www/cn/zh/customer-spotlight/cases/accelerate-ai-workload-with-amx.html

英特尔® 软件防护扩展（英特尔® SGX）

通过数据协同，引入更多源、多维、高质量的数据来打破数据孤岛，已成为各行各业深入开展大数据和 AI 应用，充分挖掘数据价值，进而加速推进数字化和智能化转型进程的共识。如今，对存储和传输状态下的数据进行加密处理已是行业的标准做法。然而，企业在数据保护方面的薄弱之处却是数据在处理器或内存中处于使用状态时。在这种情况下，个人可识别信息、电子病历和金融交易等敏感数据存在较高的被利用风险、很容易发生意外暴露或违反合规要求。

英特尔® 至强® 可扩展处理器的内置安全技术为各种数据（包括敏感、保密和处于监管之下的数据）保驾护航，使其可用于分析，进而帮助企业加速创新步伐。英特尔® SGX 是英特尔的独有技术，能够从硬件层面帮助保护使用中的数据。使用英特尔® 至强® 可扩展处理器的企业不必从数据分析和 AI 模型中剔除敏感数据，而是可通过英特尔® SGX 创建访问受限的数据安全 “飞地”。这样的隔离环境可帮助企业在更好地保护敏感数据始终处于保密状态的前提下，充分发挥其价值。

英特尔® SGX 作为先进的安全机制，可与现有基础设施一同使用，更好地保护敏感型工作负载或服务。通过使用英特尔® SGX，应用可以把代码和数据隔离在安全 “飞地” 中加以保护。至强® 处理器在高达 TB 级的内存空间中管理这些飞地。经过配置后，同一系统，甚至同一 CPU 内核上运行的其他进程将无法访问该飞地中的数据和代码，即便是具有 “根” 访问权限的进程也是如此。此外，英特尔® SGX 还解决了可信远程计算的一个基本问题：如果数据所有者想要通过某个进程来处理数据，但不能或者不想直接控制该进程，就只能信任并依赖该进程的所有者。而在英特尔® SGX 中，远程认证服务器会使用哈希值来验证 “飞地” 中的代码与开发人员发布的原始代码是否匹配，并且能够检测并阻止在飞地中植入操纵代码的企图。

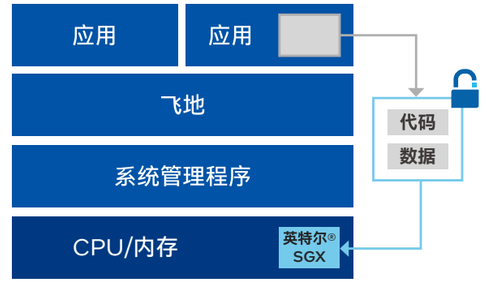


图 3-4 英特尔® SGX 通过将敏感数据隔离在容量高达 1 TB 的飞地中，帮助保护敏感数据

英特尔® SGX 经过广泛部署和研究，是数据中心可信执行环境（TEE）的重要技术实现，能够大幅减少系统内的攻击面，提供基于硬件的安全解决方案，通过专用应用隔离技术帮助保护使用中的数据。开发人员可以通过保护选定的代码和数据不被查看或修改，在 “飞地” 内执行涉及敏感数据的操作，帮助提高应用的安全性和保护数据的机密性。由英特尔® SGX 提供支持的机密计算可实现应用层面、虚拟机（VM）、容器和功能层面的数据隔离。无论是在云端、边缘还是本地环境，客户都能确保自身的计算与数据始终获得私密性和安全性更高的保护，不会暴露给云服务提供商、未经授权的管理员和操作系统，甚至是特权应用。

访问链接了解更多英特尔® SGX 的详情
https://www.intel.cn/content/dam/www/central-libraries/cn/zh/documents/2023-02/xeon-accelerated-security-product-brief-v3-q123-chinesesimplified.pdf

扫码了解更多英特尔® SGX 详情

英特尔® SST

作为目前企业级计算 CPU 代表选手的英特尔® 至强® 可扩展处理器产品家族，除了对微架构、核心数量与性能、内存通道数量和速度，以及 I/O 性能等方面持续改良以巩固优势外，它在过去数年来，还应日趋复杂的应用需求，为用户提供了越来越多面向特定场景和负载的创新技术特性。

英特尔® SST (Intel® Speed Select Technology) 能为企业多样化、差异化应用需求提供更优支持，其独特之处在于能让 CPU 根据不同应用场景或应用负载的特点及其对算力的特定要求，对处理器单个及多个核心的运行状态、频率和功耗进行精细化控制，从而能在保障更优能效的前提下满足不同负载的差异化需求。

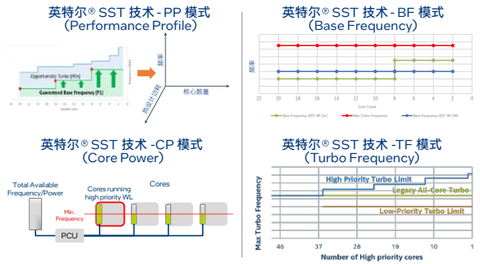


图 3-5 英特尔® SST 现有的四种模式

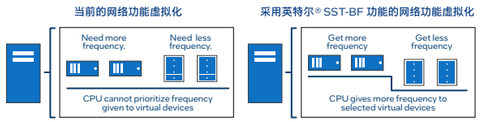
英特尔® SST- 性能配置文件 (PP)：通过配置 CPU 来适应不断变化的工作负载

英特尔® SST-PP 是英特尔® SST 家族的首款出色功能。客户可以配置一台（而不是多台）服务器来满足不断变化的工作负载处理需求，从而提高服务器利用率，降低鉴定成本。只要有一台灵活的服务器和多种配置，优化的总拥有成本是水到渠成的结果。

采用英特尔® SST-PP 的 CSP 基础设施。一台服务器、多种配置

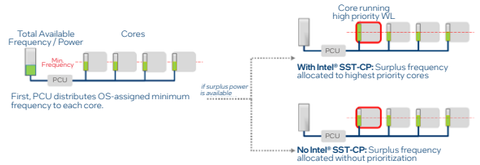
英特尔® SST- 基频 (BF)：随时随地发挥强劲动力

使用英特尔® SST-BF 功能，可以控制和分配基频，从而在关键时刻为关键工作负载提供强劲动力。如果可以对性能进行精细分配，就可以提高整体性能。



英特尔® SST-CP 模式

可对多个核心进行分组，并通过赋予各组不同的频率状态，来应对不同优先级的负载的需求。在处理器负载非常高时，SST-CP 模式会对执行低优先级任务的核心组进行降频，以确保执行高优先级任务的核心组不受影响。



访问链接了解更多英特尔® SST 的详情
https://networkbuilders.intel.com/solutionslibrary/intel-speed-select-technology-intel-sst-performance-enhancements-for-3rd-gen-intel-xeon-scalable-processor-technology-guide

扫码了解更多英特尔® SST 详情

英特尔® oneAPI 工具套件

英特尔® oneAPI 工具套件是基于新一代标准的英特尔软件开发工具，用于跨各种架构构建和部署以数据为中心的高性能应用程序。它能够通过充分利用一流的硬件特性加速计算进程，并全面兼容现有的编程模型和代码库，可确保开发者已经编写的应用能够在 oneAPI 上无缝运行。此外，开发者只需一个代码库，便可以将应用轻松迁移到新系统和加速器上，大幅缩短了迁移时间，减轻了迁移工作量。

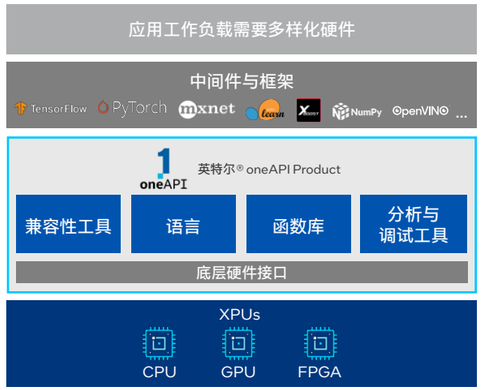


图 3-6 英特尔® oneAPI 工具套件架构

通过英特尔® oneAPI 工具套件，开发者能够使用一种通用、开放且基于行业标准的编程模型访问英特尔® CPU/GPU/FPGA。这不仅能够释放底层硬件的性能潜力，同时能降低软件开发和维护成本，并且在部署加速计算方面，英特尔® oneAPI 工具套件与专用的、受限于特定厂商的方案相比风险更低。

英特尔® oneAPI 工具套件充分利用了先进的硬件性能和指令，如用于 CPU 的英特尔® AVX-512 和英特尔® DL Boost，以及 XPU 独有的功能。英特尔® oneAPI 工具套件基于经受过长久考验的英特尔开发者工具，为开发者提供熟悉的编程语言和标准，同时与现有代码保持完全的连续性，其包括英特尔® oneAPI Base 工具包、英特尔® oneAPI AI Analytics 工具包、英特尔® oneAPI HPC 工具包及 OpenVINO™ 工具套件等不同工具。

英特尔® oneAPI AI Analytics 工具包

英特尔® oneAPI AI Analytics 工具包为数据科学家、人工智能开发人员和研究人员提供了熟悉的 Python 工具和框架，利用面向英特尔® 架构优化的库加速端到端从人工智能和数据分析流水线，实现基于面向英特尔® 架构优化的深度学习框架和工具提升训练和推理性能，并使用计算密集型 Python 包为数据分析和机器学习工作流提供落地加速。

访问链接了解更多英特尔® oneAPI 工具套件详情
https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html

扫码了解更多英特尔® oneAPI 工具套件详情

英特尔® 数据分析加速库（ oneDAL ）

英特尔为行业用户部署机器学习，也推出了一套高性能系统化方案，涵盖处理器、经优化的软件和开发人员支持，以及强大的生态系统等丰富资源。

机器学习需要强劲的计算能力。英特尔® 至强® 处理器提供了一个可扩展的基准，专门用于满足机器学习所特有的高度并行工作负载，及其对内存和架构（网络）的需求。

此外，英特尔还提供了软件支持。英特尔® oneAPI 数据分析库（oneDAL）是一套旨在帮助数据科学家和分析师们快速建立从数据预处理，到数据特征工程、数据建模和部署的端到端软件方案。它提供了建立机器学习和分析所需的各种数据分析及算法所需的高性能构建模块。目前已经支持线性回归、逻辑回归、LASSO、AdaBoost、贝叶斯分类器、支撑向量机、K 近邻、Kmeans 聚类、DBSCAN 聚类、各种决策树、随机森林、Gradient Boosting 等经典机器学习算法。这些算法经过高度优化，可在英特尔® 处理器上实现高性能。

为了开发人员在基于英特尔环境中的机器学习应用中更加方便地使用英特尔® 数据分析库 oneDAL，英特尔开源了整个项目：https://github.com/oneapi-src/onedal，并针对不同的大数据使用场景，提供全内存的、流式的和分布式的算法支持。比如 oneDAL Kmeans 可以很好地和 Spark 结合，在 Spark 集群上进行多节点聚类。另外，英特尔® oneDAL 提供了 C++ 和 Python 接口。

访问链接了解更多英特尔® oneDAL 详情
https://www.intel.com/content/www/us/en/developer/tools/oneapi/onedal.html

扫码了解更多英特尔® oneDAL 详情

英特尔® oneAPI 数学内核库（ oneMKL ）

oneMKL 是高度优化、快速、完整的数学函数库，常用于科学、工程和金融应用。oneMKL 定义了一套用于科学计算和其他应用的基本数学程序。作为 oneAPI 的一部分，oneMKL 允许在包括 CPU、GPU、FPGA 和其他加速器等各种计算设备上运行。

oneMKL 能够加快数学处理程序，提高应用性能，并减少开发时间，其具备如下特点：

- 增强的数学程序使开发人员和数据科学家能够创建高性能的科学、工程或金融应用程序；
- 核心功能包括 BLAS、LAPACK、稀疏求解器、快速傅里叶变换（FFT）、随机数生成器功能（RNG）、汇总统计、数据拟合和矢量数学；
- 针对下一代 CPU 和 GPU 进行了额外矩阵乘法优化；同时，增加了 CUDA 库函数 API 对 BLAS、LAPACK、稀疏 BLAS、向量数学、汇总统计、样条等的兼容覆盖，简化了代码向 oneAPI 和英特尔® GPU 的迁移。
- 支持第四代英特尔® 至强® 可扩展处理器的英特尔® AMX bfloat16 数据类型和英英特尔® AVX-512 bfloat16 数据类型。
- 对于以前的英特尔® 数学内核库（Intel® MKL）用户来说，是一种无缝升级。

访问链接了解更多英特尔® oneMKL 详情
https://www.intel.cn/content/www/cn/zh/developer/tools/oneapi/onemkl.html

扫码了解更多英特尔® oneMKL 详情

英特尔® 深度神经网络库（oneDNN）

英特尔® oneAPI 深度神经网络库（oneDNN）是一款面向深度学习应用的开源跨平台性能增强库，也是英特尔为了帮助开发人员充分利用英特尔® 架构，推进深度学习的研究和应用而创建的基础库。

作为开源跨平台性能库，oneDNN 针对英特尔® 架构处理器、英特尔显卡和基于 ARM 64 位架构的处理器进行了优化。oneDNN 包含了高度矢量化和线程化的构建模块，支持利用 C 和 C++ 接口实现深度神经网络，具备广泛的深度学习研究、开发和应用生态系统。目前已支持 TensorFlow、 PyTorch、MXNet、PaddlePaddle、BigDL、OpenVINO™ 工具套件等丰富的深度学习软件产品。

为了有效提升深度学习模型在基于英特尔® 架构的基础设施上的运行速度，以及提升各类神经网络中其他性能敏感型应用的效率，oneDNN 提供了众多优化的深度学习运行和操作基元，可应用于不同的深度学习框架，以确保通用构建模块的高效实施。

oneDNN 目前已成为众多深度学习框架在 CPU 上运行时的基本配置，开发者可在深度学习框架的安装和应用中，直接获取 oneDNN 带来的性能提升。

访问链接了解更多英特尔® oneDNN 工具套件详情
主页：https://www.intel.com/content/www/us/en/developer/tools/oneapi/onednn.html
代码：https://github.com/oneapi-src/oneDNN

扫码了解更多英特尔® oneDNN 详情

面向英特尔® 架构优化的深度学习框架

面向英特尔® 架构优化的 TensorFlow，通过计算图、内存池分配器与多个线程库等组件的优化，能够确保深度学习工作负载在各种情况下都可利用英特尔® MKL-DNN 基本运算单元高效运行。

英特尔® Python 分发版提供了编写 Python 原生扩展所需的一切，如 C++ 和 Fortran 编译器、数学库和分析器，并且集成 NumPy、SciPy scikit-learn、pandas、Jupyter、matplotlib、mpi4py 等多个高性能数据分析和数学库，能够满足计算密集型应用需求。

面向英特尔® 架构优化的 Caffe，集成了英特尔® 数学核心函数库，专门面向高级矢量扩展指令集英特尔® AVX2 和英特尔® AVX-512 做了优化，且具备更多处理器优化功能，展现了更优的性能，并支持多节点分布程序训练。

英特尔开源的统一大数据和人工智能平台 BigDL 可以无缝、直接运行在现有的 Apache Spark 和 Hadoop 集群之上，是在处理器平台上实现大数据分析 +AI 应用的关键。BigDL 支持 PyTorch，TensorFlow，OpenVINO 等主流 AI 应用框架，可以将用户程序从笔记本无缝扩展到大数据集群上。

面向英特尔®架构优化的 TensorFlow 扩展包（ITEX）

面向英特尔® 架构优化的 TensorFlow 扩展包（Intel® Extension for TensorFlow，ITEX），是基于 TensorFlow Pluggable Device 接口的异构高性能深度学习扩展插件，可将英特尔® XPU（GPU、CPU 等）设备引入 TensorFlow 开放 AI 工作负载加速的开源社区。为了显著提升性能，英特尔持续采用多种措施对 TensorFlow 进行优化：

- **运算符优化**：针对 CPU 中的运算符进行优化，并通过英特尔® oneAPI DPC++ 编译器实现所有 GPU 运算符优化。用户无需任何额外设置即可默认获得这些算子优化收益，此外，还开发了多个用于使用 itex.ops 命名空间提升性能的定制运算符，以扩展 TensorFlow 公共 API 实现以获得更好的性能。
- **计算图优化**：将指定的运算模式融合到新的单一运算中，以获得更好的性能，例如 Conv2D+ReLU、Linear+ReLU 等。混合精度使用较低精度的数据类型（FP16 或 BF16），使模型在训练和推理过程中运行速度更快，内存消耗更少。
- **低精度优化**：TensorFlow 开发者可以通过低精度数据类型 Bfloat16 为训练与推理模型加速，获得至多 2 倍的性能提升且保持模型精度不变。ITEX 提供了包括手动、自动等多种 Bfloat16 模型转换解决方案，用户可以根据实际情况选择不同的方案来满足对于性能或使用体验的要求。

访问链接了解更多 TensorFlow 扩展包详情
主页：https://www.intel.com/content/www/us/en/developer/tools/oneapi/optimization-for-tensorflow.html
性能数据：https://github.com/intel/intel-extension-for-tensorflow

扫码了解更多 TensorFlow 扩展包详情

OpenVINO™ 工具套件

OpenVINO™ 工具套件是英特尔推出的一款加速深度学习推理及部署的软件工具套件，用以加快高性能计算机视觉处理和应用。该工具允许异构执行，支持 Windows 与 Linux 系统，以及 Python/C++ 语言，能够有效推进计算机视觉技术在从智能摄像头、视频监控、机器人，到智能交通、智能医疗等领域的深入应用。

本工具套件提高了计算机视觉解决方案的性能，缩短了开发时间，简化了从英特尔提供的丰富硬件选项中获得效益的途径，而这些选项可以提高性能、降低功耗并最大化硬件利用率——让用户可以低资源获得高收益，并为新的产品设计提供个性化空间。

通过基于深度卷积神经网络（CNN），扩展英特尔硬件（包括加速器）的工作负载，使得 OpenVINO™ 工具套件可依托基于英特尔® 架构的 CPU 和 GPU 来增强视觉系统的功能和性能。最新发布的 OpenVINO™ 版本已能支持第四代英特尔® 至强® 可扩展处理器，并通过英特尔® AMX、英特尔® AVX-512 以及采用 VNNI 的英特尔® DL Boost 技术来提升推理性能，可帮助客户在不改变软件的基础上，快速完成硬件产品升级和算法移植，从而助其在边缘侧快速实现高性能计算机视觉与深度学习应用的开发：

- 释放 CNN-based 的网络在边缘设备的性能瓶颈
- 基于通用 API 接口在基于英特尔 ® 架构的 CPU 和 GPU 上运行

基于英特尔平台优化的 OpenVINO ™ 工具套件主要包括模型优化器（Model Optimizer）和推理引擎（Inference Engine）两个核心组件：

■ **模型优化器：**

- 模型优化器是一种跨平台命令行工具，它可将训练后的网络模型从其源框架转换为开源、且与 nGraph 兼容的 IR（中间表示）文件以用于推理操作，IR 文件是由 bin（经训练的数据文件）和 xml（描述网络拓扑的文件）两种格式文件组成。一方面，模型优化器可以在导入由 Caffe、TensorFlow、MXNet、PyTorch、Keras 以及 ONNX 等流行框架训练好的模型后，执行相应优化，包括去除多余的层，并在可能的情况下将操作分组为更简单、更快的图（Graph）等。
- 另一方面，OpenVINO™ 工具套件也可以进行模型量化过程。PyTorch 等流行框架中训练的模型通常为 FP32 精度数据格式，而英特尔® 至强® 可扩展处理器等计算平台已经支持 INT8 等低精度数据格式下的模型推理，其可在损失很小精度的前提下实现更高的推理效率。量化过程是将基于高精度数据格式的模型（由 IR 文件表示）转为低精度，并加入校准过程以保证精度不受损失。

■ **推理引擎：**

- 支持硬件指令集层面的深度学习模型加速运行，支持的硬件设备主要包括：英特尔® CPU 和 GPU。实现用户一次开发，即可面向不同平台部署并获得一致的性能表现，有效提升 AI 开发与部署效率。

OpenVINO™ 工具套件在英特尔平台上让视觉成为现实，已帮助众多用户轻松开发和快速部署计算机视觉应用程序，在多种深度学习应用场景展示了人工智能解决方案所蕴藏的巨大潜力。

访问链接了解更多 OpenVINO™ 详情
主页：https://www.intel.cn/content/www/cn/zh/developer/tools/openvino-toolkit/overview.html
代码：https://github.com/openvinotoolkit/openvino
性能数据：https://docs.openvino.ai/latest/openvino_docs_performance_benchmarks.html

扫码了解更多 OpenVINO™ 详情

本手册涉及的专业词汇表

英文全称	英文缩写	中文全称
Algorithmic Concepts		算法概念
Application Programming Interface	API	编程接口
Automatically Tuned Linear Algebra Software	ATLAS	自动调优线性代数系统
Basic Linear Algebra Subroutine	BLAS	基本线性代数子程序
Batch Size		批处理参数
Cardiac Magnetic Resonance	CMR	磁共振成像
Chain of Thought	CoT	思维链
Clinical Decision Support System	CDSS	临床决策支持系统
Clinical Information System	CIS	临床信息系统
Command Line Interface	CLI	便捷命令
Computed Tomography	CT	电子计算机断层扫描
Computational Motifs		计算模式
Concat Ops		连接操作
Consensus model		共识模型
Constant Folding		常量折叠
Convolution Ops		卷积操作
Convolutional Layer		卷积层
Convolutional Neural Network	CNN	卷积神经网络
Cosine Similarity		余弦相似度
Deep Supervision		深度监督
Deeping Learning	DL	深度学习
Double Data Rate	DDR	双倍速率
	DNA	脱氧核糖核酸
Dynamic Random-Access Memory	DRAM	动态随机存取存储器
Electronic Medical Record	EMR	电子病历
Euclidean Distance		欧氏距离
Feature extraction		特征抽取
Feature map		特征地图
Fully Connected Layer		全连接层
Fully Convolutional Network	FCN	全卷积网络
Fused Multiply Add	FMA	宽融合乘加
General matrix multiply	GEMM	通用矩阵乘法
Genome-Wide Association Studies	GWAS	全基因组关联分
Geometric pattern		几何特征

英文全称	英文缩写	中文全称
GNU Compiler Collection	GCC	GNU 编译器套件
High Content screening	HCS	高内涵筛选
Hidden Markov Model	HMM	隐马尔可夫模型
Hospital Information System	HIS	医院信息管理系统
Intel® Advanced Matrix Extensions	Intel® AMX	英特尔 ® 高级矩阵扩展
Intel® Advanced Vector Extensions	Intel® AVX	英特尔 ® 高级矢量扩展指令集
Intel® Deep Learning Deployment Toolkit	Intel® DLDT	英特尔 ® 深度学习部署工具
Intel® Extensions for PyTorch	IPEX	面向 PyTorch 的英特尔 ® 扩展优化框架
Intel® Streaming SIMD Extensions	Intel® SSE	英特尔 ® 流式单指令
Intel® Ultra Path Interconnect	Intel® UPI	英特尔 ® 超级通道互联
Last Level Cache	LLC	末级高速缓存
Layer Fusion		层融合
Layer-wise Relevance Propagation	LRP	层级相关性传播
Learning Rate	LR	学习率
Liquid-Based Cytologic Preparation	LBP	宫颈液基细胞学制片
Large Language Model	LLM	大语言模型
Likelihood Ratio	LLR	似然比
Logistics Regression	LR	逻辑回归
Magnetic Resonance Imaging	MRI	核磁共振成像
Mahalanobis distance		马氏距离
Math Kernel Library for Deep Neural Networks	MKL-DNN	数学核心函数库
Matrix multiplication		矩阵乘法
Multi-scale Convolutional Neural Networks	M-CNN	多尺度卷积神经网络
Multi-Scale Prediction		多尺度预测
Multiple sequence alignment	MSA	多序列比对
Non- Uniform Memory Access Architecture	NUMA	非统一内存访问架构
Open Message Passing Interface	OpenMPI	开放消息传递接口
Open Multi-Processing	OpenMP	
Open Neural Network Exchange	ONNX	开放神经网络交换
Operations Per Second	OPS	每秒操作数
Optical Character Recognition	OCR	光学字符识别
Picture Archiving and Communication Systems	PACS	影像归档和通信系统
Pixel Intensity		像素亮度
Platform as a Service	PaaS	平台即服务

英文全称	英文缩写	中文全称
Pooling Layer		池化层
Position-Sensitive RoI Pooling		敏感的 ROI 池化操作
Position-sensitive score map		位置敏感得分映射
Positron Emission Tomography CT	PET-CT	正电子发射计算机断层显像
Primitive		多种原语
Principal component analysis	PCA	基于主成分分析
Programming Paradigm		编程范式
Region of Interest	ROI	兴趣区域
Region Proposal Network	RPN	区域生成网络
Reorder Ops		重排序操作
Resample Ops		重采样
Residual Net	ResNet	残差网络
Root Mean Squared Error	RMSE	均方根误差
	RNA	脱氧核糖核酸
Secure Federated Learning	SFL	安全联邦学习
Single Instruction Multiple Data (SIMD)	SIMD	单指令多数据流
Single Nucleotide Polymorphisms	SNPs	单核苷酸多态性
Sliding Window Algorithm		滑动窗口算法
Software as a service	SaaS	软件即服务
Standard Uptake Value	SUV	标准化摄取值
Standardized Euclidean distance		标准化欧氏距离
Support Vector Machines	SVM	支持向量机
Tensor Processing Primitives	TPP	张量计算原语
Tile Matrix Multiply Unit	TMUL	Tile 矩阵乘法
Total Cost of Ownership (TCO)	TCO	总拥有成本

扫码访问英特尔官网
了解更多英特尔在人工智能
领域的技术实践

关注英特尔数据中心微信公众号
随时了解最新活动与资讯

扫码下载
《英特尔中国医疗健康行业
AI实战手册》

免责声明：

性能测试中使用的软件和工作负荷可能仅在英特尔微处理器上进行了性能优化。诸如 SYSmark 和 MobileMark 等测试均系基于特定计算机系统、硬件、软件、操作系统及功能。上述任何要素的变动都有可能导致测试结果的变化。请参考其他信息及性能测试（包括结合其他产品使用时的运行性能）以对目标产品进行全面评估。更多信息，详见 www.intel.com/benchmarks。

在特定系统的特殊测试中测试组件性能。硬件、软件或配置的差异将影响实际性能。当您考虑采购时，请查阅其他信息来源评估性能。关于性能和基准测试程序结果的更多信息，请访问 www.intel.com/benchmarks。

英特尔技术特性和优势取决于系统配置，并可能需要支持的硬件、软件或服务得以激活。产品性能会基于系统配置有所变化。没有任何产品或组件是绝对安全的。更多信息请从原始设备制造商或零售商处获得，或请见 intel.com。

优化声明：英特尔编译器针对英特尔微处理器的优化程度可能与针对非英特尔微处理器的优化程度不同。这些优化包括 SSE2、SSE3 和 SSSE3 指令集和其他优化。对于非英特尔微处理器上的任何优化是否存在、其功能或效力，英特尔不做任何保证。本产品中取决于微处理器的优化是针对英特尔微处理器。不具体针对英特尔微架构的特定优化为英特尔微处理器保留。请参考适用的产品用户与参考指南，获取有关本声明中具体指令集的更多信息。

没有任何产品或组件是绝对安全的。

描述的成本降低情景均旨在在特定情况和配置中举例说明特定英特尔产品如何影响未来成本并提供成本节约。情况均不同。英特尔不保证任何成本或成本降低。

英特尔并不控制或审计第三方数据。请您审查该内容，咨询其他来源，并确认提及数据是否准确。