生成技术在人工智能平台中的应用探索

夏正勋，杨一帆，罗圣美，赵大超，张燕，唐剑飞
星环信息科技（上海）有限公司，上海 200233

摘要
随着人工智能（AI）技术的发展，AI应用进入了快速普及期，面对快速增长的市场需求，AI平台有必要引入自动化方法提升开发效率。在分析生成技术研究进展、AI平台现状及挑战的基础上，基于生成技术实现了AI平台的前后端适配、性能优化、模型安全提升等核心工作的自动化。新方法可以根据上下文的需要，生成数据或代码，以一种更灵活的方式满足AI应用及内核优化的需求，避免了大量的手工工作，有效提升了开发效率，降低了开发成本。

关键词
生成技术；人工智能；自动化

中图分类号：TP311.13　　文献标识码：A　　doi: 10.11959/j.issn.2096-271.2020058

Application and exploration of automatic generation technology in artificial intelligence platform

XIA Zhengxun, YANG Yifan, LUO Shengmei, ZHAO Dachao, ZHANG Yan, TANG Jianfei
Transwarp Information Technology (Shanghai) Co., Ltd., Shanghai 200233, China

Abstract
With the development of artificial intelligence (AI) technology, AI applications have entered a period of rapid popularization, facing the rapidly growing market demand, it is necessary for AI platforms to use automated methods to improve development efficiency. Based on the analysis of the research progress of generation technology, the status quo and challenges of AI platform, based on generation technology, the automation of AI platform's front and rear end adaptation, performance optimization, and model security enhancement were realized, which can generate data or code according to the needs of the context and to meet the requirements in a more flexible way. It can also avoid a lot of manual work and can effectively improve development efficiency and reduce development cost.

Key words
generation technology, artificial intelligence, automation

1 引言

随着人工智能(artificial intelligence, AI)技术的快速发展,特别是在深度学习(deep learning, DL)技术的推动下,人工智能的应用需求呈爆发式增长。AI平台是AI功能的载体,多样化的应用需求对AI平台提出了越来越高的要求,在不同的发展时期,AI平台有不同的关注点。在发展初期,AI平台关注其基础能力实现,如对训练及推理的支持能力、支持的算法种类等。在应用普及期,AI平台关注其落地能力,如性能优化、可视化管理、虚拟化支持等。近年来,AI进入快速推广期,AI平台更关注其商用成本及对创新特性的支持能力,如AI硬件支持种类、数据安全特性支持等。

为了满足不同阶段的不同需求,AI平台需要不断优化升级,增强功能,这导致AI平台处理流程越来越复杂,增加了AI平台优化改造的难度及工作量。为此,需要一种更灵活的AI平台内核实现手段支持新功能的开发,而生成技术可以根据上下文的需要生成数据或代码,以一种更灵活的方式满足AI上层应用的需求及内核自身的改进需求,提高AI平台的灵活性及稳定性,快速实现AI平台的自我优化。

2 生成技术的当前研究方向与现状

生成技术可以根据上下文的需要生成符合特定规则的内容(如代码、数据等),具体包含代码生成技术、参数空间生成技术、数据样本生成技术等。

代码生成技术[1]应用于GNU编译器套件(GNU compiler collection, GCC)、低级虚拟机(low level virtual machine, LLVM)、Clang(C language family frontend for LLVM)等编译器中。代码生成模块作为编译器前端(frontend)的一部分,从语法和词法分析处理模块获得抽象语法树,并向编译器后端(backend)提供字节码,是连接编译器前端和后端的纽带,该过程如图1所示。

受益于代码生成技术,编译器可以将不同的编程语言(如C、C++、Java等)的源码输出为统一的中间表示(intermediate representation, IR),并针对中间表示进行统一优化,提升代码执行效率。另外,代码生成部分也为不同的编译器后端提供标准化的字节码输入,使编译器能够在不改变源代码的情况下,实现同一份源码支持多个编译器后端(如x86硬件平台后端、PowerPC硬件平台后端、ARM硬件平台后端等),使程序具备跨平台的支持能力。近些年,随着深度学习技术的发展,出现了众多深度学习框架,如Caffe、TensorFlow、PyTorch、MXNet等,不同深度学习框架输出的模型相互之间并不兼容。为解决这一问题,华盛顿大学计算机科学与工程学院于2016年发布了NNVM(neural network virtual machine)[2]编译器,NNVM借鉴了LLVM的思想,通过代码生成技术为不同的深度学习框架模型提供统一的深度学习中间表示(deep learning intermediate representation, DLIR)语言,不需要编码即可支持多种深度学习框架模型跨硬件平台的推理执行。

参数空间生成技术主要应用于AI算法的超参数自动调优阶段。在AI的上下文中,超参数需要在开始学习过程之前进行设置,而不是通过训练得到参数数据。通常情况下,超参数主要依据工程师的经验配置,当参数数量增多时,参数组合情况

倍增，人工配置难以取得很好的效果，因此超参数自动化调优技术的出现减轻了AI工程师的负担，使其将工作重心从烦琐、重复的选型和调参任务转移到数据分析上。超参数自动化调优技术通常包含参数空间的生成与参数空间的优化选择[3]两个阶段，参数空间的生成是从理想状态下的所有参数组合中选择有潜力的候选配置，参数空间可以基于规则方法生成或基于元学习技术方法[4]生成。参数空间的优化选择阶段可以采用的方法有基于强化学习的超参数优化方法[5]、基于改进粒子群算法的深度学习超参数优化方法[6]、基于贝叶斯新型深度学习超参数优化方法[7]等。

数据样本生成技术通常指自主学习原始样本的分布规律，生成新的数据样本，例如目标检测场景中数据集的半自动生成[8]、基于生成对抗网络（generative adversarial network，GAN）的小样本数据的生成[9]等。在AI安全领域，对抗样本的生成技术指在原有样本的像素上添加扰动的方法[10]，使包括卷积神经网络在内的深度学习模型的准确率显著降低。数据样本生成技术的应用丰富了AI训练数据集，解决了训练样本数量和样本多样性不足的问题，有效地提升了模型的精度及鲁棒性。

3 生成技术在AI平台中的应用及系统实现

AI平台是提供“算法、算力、数据”基础能力的平台，在AI平台之上是AI的各类行业应用。艾瑞咨询发布的《2019年中国人工智能产业研究报告》将AI的服务分为基础层服务、技术层服务、产品与解决方案服务，AI平台主要涵盖基础层服务及技术层服务。具体而言，基础层服务主要包含

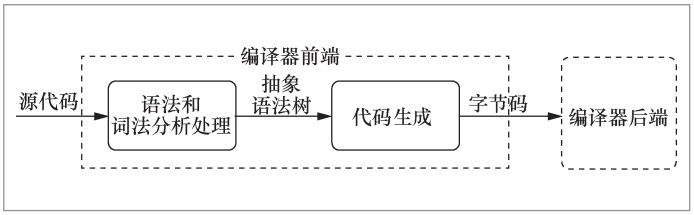

图1 代码生成在编译器处理流程中的位置

AI芯片、AI框架、AI边缘设备、AI容器云服务、AI数据服务等；技术层服务主要包含计算机视觉、语音识别、自然语言处理、知识图谱、机器学习等算法及模型服务等。AI平台的建立有助于降低技术门槛，让所有人都能享受到AI技术进步带来的红利。

但AI技术的快速发展及其相关应用的快速普及也为AI平台带来了新的挑战。无论是新型算法、新型硬件的支持还是更高的AI安全可靠性要求，均需要从AI的基础层、技术层进行创新，而这不可避免地会对AI平台的原有架构、流程及功能做出变更。依靠传统人工编码的方式支持AI平台新特性的开发，工作量大，开发周期长，对新需求的响应速度较慢。本文对生成技术在AI平台进行应用实践及探索思考，期望能够为AI平台的架构设计及技术实现提供一种新的思路，快速响应内外部需求的柔性扩展。本文将生成技术应用于AI平台的模型支持、运行时（runtime）等核心模块，包括自动化前后端适配、自动化调优、自动对抗学习等功能模块，从而可以根据上下文的需要，自动地生成数据或代码，避免了大量的手工工作，有效地提升了AI平台的开发效率，降低了开发成本。

3.1 基于代码生成技术的自动化前后端适配

代码生成技术是一种利用程序生成代码的技术，与人工编写代码相比，代码生

成技术有效解决了人工编写代码工作量大、耗时长的问题，提高了软件开发效率。近些年，随着AI软硬件的快速演进发展，特别是国产软硬件的发展，为了能够实现AI平台对各类深度学习框架（如TensorFlow、MXNet、PyTorch、PaddlePaddle、MindSpore等前端框架）的广泛兼容，同时实现对后端AI硬件的广泛支持，基于代码生成技术实现了自动化前后端适配子系统，自动化前后端适配流程如**图2**所示。

首先，针对不同的学习框架，自动化前后端适配子系统提供不同的解析脚本对模型进行解析，以提取模型中的网络结构定义、网络参数及超参数等信息。统一IR是自动化前后端适配子系统定义的中间表示，统一IR考虑了所有深度学习框架模型的算子支持情况，统一IR可以与模型中的算子一一对应，对应关系被预定义在算子匹配规则表中。算子匹配规则表示例如**图3**所示，其中冒号前为模型中的算子，冒号后为统一IR中的算子。

自动化前后端适配子系统在遍历神经网络的过程中，使用代码生成技术，根据算子匹配规则表的匹配关系，生成统一的计算图，如**图4**所示。

自动化前后端适配子系统可针对计算图进行图级别的优化，这些优化包含重复子句消除、计算简化、卷积计算核合并、计算节点合并等。自动化前后端适配子系统根据不同的目标硬件平台选择不同的硬件runtime，同样基于代码生成技术将统一计算图生成为特定硬件的执行程序。如果目标硬件为NVIDIA GPU，则选择的runtime为统一计算架构（compute unified device architecture，CUDA）；如果目标硬件为AMD GPU，则选择的runtime为RCOM。接着就可以调用与硬件对应的编译器对执行代码进行优化、编译，最终生成可以在不同硬件平台上运行的可执行模块。

综上，基于代码生成技术的应用，依据简单的前后端类型配置信息，自动化前后端适配子系统可以自动地将不同深度学习框架模型转化为在特定硬件上的可执行代码，减少了大量的模型转换、硬件适配工

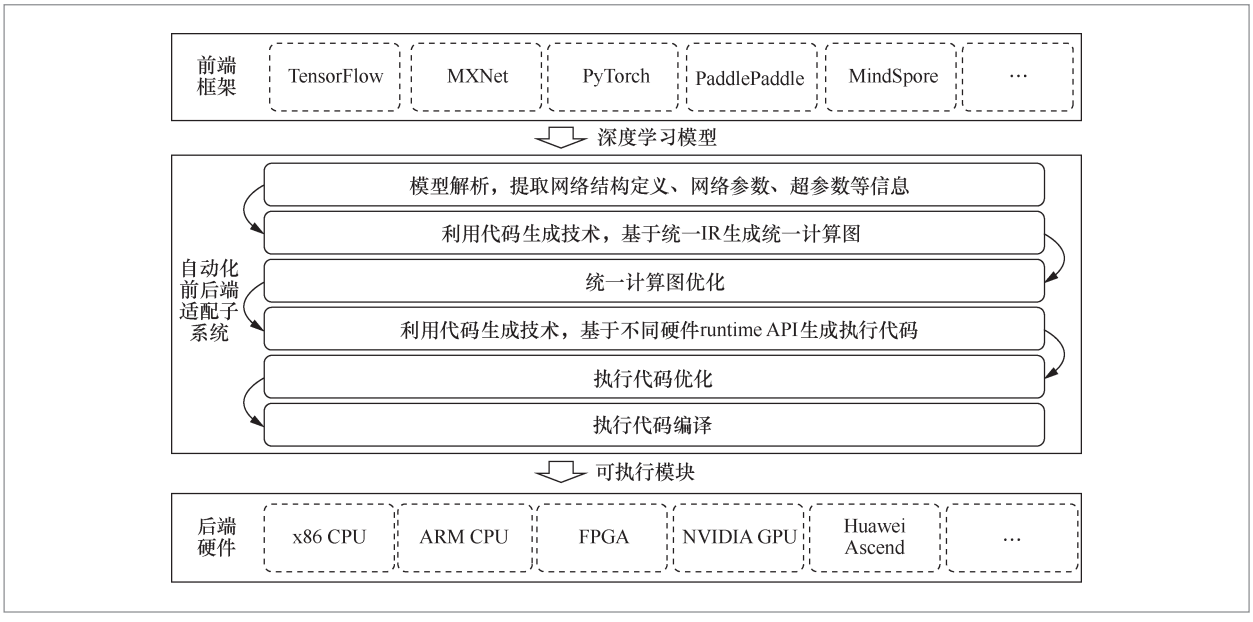

图 2 自动化前后端适配流程

作，提升了AI平台的开发效率及易用性。

3.2 基于参数空间生成技术的自动化调优

超参数是AI模型中的框架参数，如聚类算法中的类别数目、矩阵乘法中的数据形状的定义等。超参数与训练过程中学习到的权重参数不一样，其通常由人工设置，不断试错调整，这往往会花费大量的时间。因此，基于参数空间生成及参数空间搜索的自动化参数优化技术可以实现超参数调优自动化，不需要人工参与，速度更快，性能更优。其核心思想是：建立一个足够大的搜索空间，保证可能的参数组合全部被包含在这个搜索空间里；快速地搜索这个空间，获取最优的参数组合，可以利用随机搜索、网格搜索、遗传算法、极端梯度提升树（extreme gradient boosting，XGBoost）[11]方法对参数空间进行检索。

下面以$\boldsymbol{A}[m,k]\times\boldsymbol{B}[k,n]=\boldsymbol{C}[m,n]$矩阵乘法为例，说明自动化调优过程中参数空间生成的应用，其计算过程如**图5**所示。

如**图5**所示，考虑到内存空间有限，对于大型的矩阵乘法，通常采用分片计算的方式。在计算过程中，参与单次计算的3个数据块$\boldsymbol{a}[m1,k1]$、$\boldsymbol{b}[k1,n1]$、$\boldsymbol{c}[m1,n1]$均能够在缓存中被连续访问，这可以有效地减少上下文切换，极大地提升计算效率。但不同的AI芯片的缓存配置不同，因此人工配置难以达到最优计算性能，需要依靠自动化方法实现最优参数搜索的工作。通常首先以2^n为基本单位对输出数据的每个维度进行分割，如$\boldsymbol{A}[512,512]\times\boldsymbol{B}[512,512]=\boldsymbol{C}[512,512]$的计算，分割后结果为[[512, 1], [256, 2], [128, 4], [64, 8], [32, 16], [16, 32], [8, 64], [4, 128], [2, 256], [1, 512]]，输出块的形状

```
'ConcatV2'                : _concatV2(),
'Conv2D'                  : _conv('conv'),
'DecodeJpeg'              : _decode_image(),
'DepthwiseConv2dNative'   : _conv('depthwise'),
'Equal'                   : _broadcast('equal'),
'Elu'                     : _elu(),
'Exp'                     : AttrCvt('exp'),
'ExpandDims'              : _expand_dims(),
'Fill'                    : _fill(),
'Floor'                   : AttrCvt('floor'),
'FusedBatchNorm'          : _fused_batch_norm(),
'FusedBatchNormV2'        : _fused_batch_norm(),
'Gather'                  : _gather(),
'GatherV2'                : _gather(),
'Greater'                 : _broadcast('greater'),
'GreaterEqual'            : _broadcast('greater_equal'),
'Identity'                : _identity(),
'LeakyRelu'               : AttrCvt('leaky_relu'),
```

图 3 算子匹配规则表示例

```
v0.0.4
fn (%DecodeJpeg/contents: Tensor[(299, 299, 3), uint8], %conv/conv2d_params:
  %0 = cast(%DecodeJpeg/contents, dtype="float32");
  %1 = expand_dims(%0, axis=0);
  %2 = image.resize(%1, size=[299, 299], layout="NHWC", align_corners=False);
  %3 = subtract(%2, meta[relay.Constant][0]);
  %4 = multiply(%3, meta[relay.Constant][1]);
  %5 = transpose(%4, axes=[0, 3, 1, 2]);
  %6 = transpose(%conv/conv2d_params, axes=[3, 2, 0, 1]);
   %7 = nn.conv2d(%5, %6, strides=[2, 2], channels=32, kernel_size=[3, 3]);
   %8 = transpose(%7, axes=[0, 2, 3, 1]);
   %9 = nn.batch_norm(%8, %conv/batchnorm/gamma, %conv/batchnorm/beta, %conv/
   %10 = %9.0;
   %11 = nn.relu(%10);
   %12 = transpose(%11, axes=[0, 3, 1, 2]);
   %13 = transpose(%conv_1/conv2d_params, axes=[3, 2, 0, 1]);
   %14 = nn.conv2d(%12, %13, channels=32, kernel_size=[3, 3]);
   %15 = transpose(%14, axes=[0, 2, 3, 1]);
   %16 = nn.batch_norm(%15, %conv_1/batchnorm/gamma, %conv_1/batchnorm/beta,
   %17 = %16.0;
```

图 4 统一计算图示例

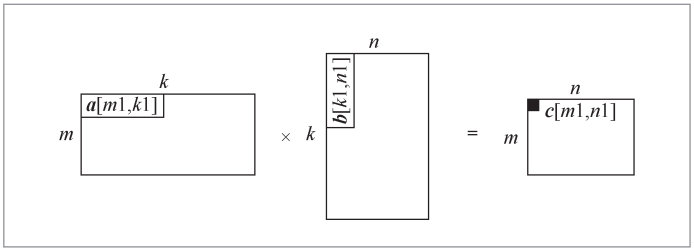

图 5 二维矩阵乘法示例

（shape）共有10种，[32,16]表示$\boldsymbol{A}$[32,k]×$\boldsymbol{B}$[k,16]=$\boldsymbol{C}$[32,16]。接着以2^n为基本单位对k轴做分割，k同样有10种取值，因此生成的参数组合空间中的参数组合数目为100种。

在具体应用中，参数空间的生成方式与参数搜索方法有关。例如，使用网格搜索方法需要生成所有参数组合，并对所有参数组合进行遍历，这并不是一种高效的参数优化方式；使用随机搜索方法，可能效果特别差，也可能效果特别好，在尝试次数与网格搜索方法的尝试次数相同的情况下，通常随机搜索方法的最值会更大，变化幅度也更大，但这不会影响最终结果。在实现随机搜索时可以进行优化，过滤可能出现过的参数组合，避免重复生成及重复计算。使用遗传算法进行参数调优时，开始可以使用随机生成方法对“种群”进行初始化工作，在优化过程中完成参数的“复制”“交叉”“变异”等处理，当尝试总次数大于参数空间总数时，“遗传”结束。使用XGBoost方法对参数空间进行搜索时，每一批计算的参数组合中的95%可以遍历生成，5%可以随机生成。另外，不同场景中的参数生成规则可能不同，因此还需要对参数生成规则做一定的管理，在插件式管理的基础上可以组合出更强的参数生成能力。

在AI平台中，与自动化参数调优子系统类似的还有自动网络设计及调优子系统，网络生成技术通过神经网络基础算子的堆叠组合，改变算子间的链接权重或拓扑结构等生成规则，构建神经网络架构空间，然后在生成的网络架构空间中使用遗传算法、XGBoost等方法完成网络架构的优化选择。

3.3 基于数据样本生成技术的自动对抗学习

随着AI技术的深入应用，人们越来越关注AI自身的安全性问题。2018年，360安全研究院发布的《AI安全风险白皮书》指出：深度学习框架中对抗机器学习的恶意样本生成、训练数据的污染等可能导致AI驱动的识别系统出现混乱，形成漏判或者误判，甚至导致系统崩溃或被劫持。Kurakin A等人[10]提出了大规模对抗机器学习系统，通过将对抗样本加入训练过程，增强模型的抗攻击能力。在分析对抗样本及数据毒化等AI攻击方法的基础上，结合对抗学习理念，可以构建商用化的AI对抗学习子系统，其系统处理流程如图6所示。

AI对抗学习子系统基于数据样本生成技术，针对特定的模型及原始样本生成能够误导模型判断的对抗样本，对抗样本的生成过程是在原样本上生成能够让模型做出误判的微小扰动的过程。具体而言，这类扰动可以通过梯度方法或仿射平面方法等白盒方法生成，如FGSM[12]、C&Wattacks[13]、DeepFool[14]等，或者通过生成网络方法、差分进化算法等黑盒方法生成，如UPSET[15]、ANGRI[15]、Houdini[16]、One-Pixel[17]等。对抗样本生成器可以使用上述方法生成对抗样本，对抗样本的特点是与正常样本偏差很小，但模型输出结果偏差很大，通过将对抗样本加入学习过程，可以提升模型的抗攻击能力。这个过程是一个自动化的持续学习过程，通过不断地生成、训练，持续提升模型的安全性。同样的思路，对于数据毒化的攻击来说，可以通过毒化样本生成器生成毒化样本，这类样本的特点是与正常样本偏差比较大，但模型输出结果偏差很小。将毒化样本加入学习过程，可以提升模型抗毒化的能力。在模型训练的过程中引入对抗攻击，从而提升模型对对抗攻击的鲁棒性是一种行之有效的提升模型安全性的方法，但理论上也存在局限性。该方法需要使用高强度的对抗样本，网络架构也需

要具有充足的表达能力，并且不能排除存在新对抗样本的可能性。

4 应用案例

AI、大数据等互联网技术和互联网企业的发展，为电网公司进行企业转型提供了鲜明的指引，电网企业需要使用新的技术手段对整体业务进行赋能。以星环信息科技（上海）有限公司的Sophon AI平台在某世界500强电力集团公司智能巡检项目中的应用为例，其应用场景如**图7**所示。

本项目涵盖了固定摄像头、无人机、直升机、机器人、移动作业、卫星遥感等空天一体化全方位巡检方式，采集数据日增量达TB级。为了支撑当前的线路智能巡检要求，在变电站一级部署了大量嵌入式边缘计算设备，边缘设备有两种型号EDGE100及EDGE200，其中EDGE100处理器为ARM Cortex-M系列处理器，EDGE200处理器为RISC-V定制处理器，操作系统均为CentOS7。原模型既有TensorFlow模型也有PyTorch模型，运行在Windows x86服务器之上，本项目中需要将AI模型从x86服务器平台迁移至嵌入式设备。迁移过程通常会遇到如下问题：

- 模型需要对软硬件环境重新进行适配；
- 嵌入式设备处理能力不足，需要优化模型，提升计算效率，使模型能够正常运行。

按通常做法，需要通过人工方式分别将不同框架模型迁移到不同硬件平台，再进行性能调优，这些工作通常需要投入大量资源，从开发到功能上线周期较长，耗时耗力。Sophon AI平台基于代码生成技术实现了自动化前后端适配功能，基于代码生成技术可以将主流深度学习框架的模型转换为统一的IR，实现跨学习框架的模型快速适配。在此基础上，同样基于代码生成技术，Sophon AI平台可将统一的IR生成适配不同硬件的执行代码，从而实现跨硬件平台的模型快速迁移。借助Sophon AI平台自动化前后端适配、自动化调优的能力，本项目中红外发热点检测、杆塔倾斜检测、绝缘子脱落检测等模型在6 h之内实现了模型迁移、部署、微调的工作，相对于传统人工迁移的方式，大大缩短实施时间。此外Sophon AI平台还基于参数空间生成技术实现了超参自动优化功能，相对于按经验配置，自动优化的参数配置更能最大化发挥EDGE100、EDGE200设备的计算性能，优化后单图片识别的平均处理时长由5.13 ms缩短至4.52 ms，推理效率平均提升了11.9%。本项目的成功实施推动

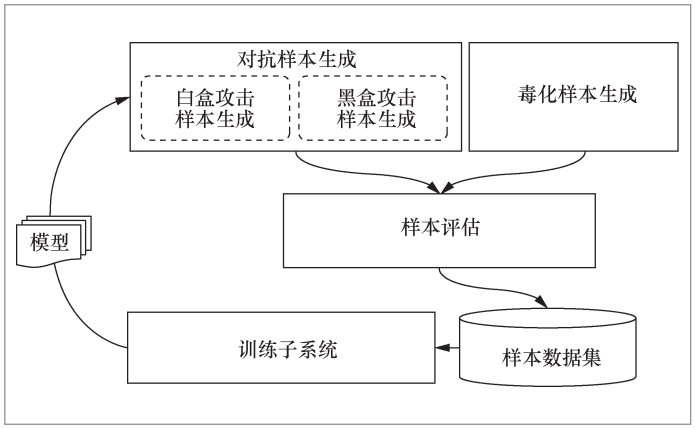

图 6 AI 对抗学习子系统流程

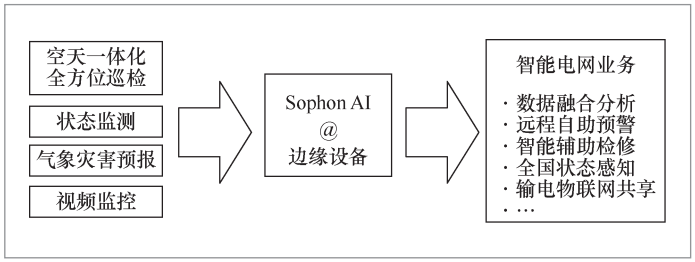

图 7 输电线路智能巡检应用场景

了AI技术在电力电网领域的应用，也验证了生成技术在AI平台中的应用价值。

5 结束语

多框架模型的支持、多硬件平台的支持、模型计算性能调优、模型安全性的提升是AI平台的核心功能，本文在AI平台基础技术实现层面进行了思考与实践，借助生成技术实现上述工作的自动化，避免了在代码迁移、适配、调优、测试等工作环节的重复投入，具有现实的意义。代码生成、参数生成、网络生成、样本生成等生成技术的应用使Sophon AI平台灵活易用，基于生成技术从前后端适配、性能调优、安全提升等多个层面打造高效的AI开发平台，避免了大量人工开发的工作，缩短了需求响应周期，全方面地提升了AI平台的应用开发效率。面向未来，从更高的要求出发，AI系统还需要具备环境自适应性、自我进化能力，而生成技术具备适配上下文的需要、动态输出合适对象的能力，这种柔性的动态生成能力相对于固化的应用功能，无疑更贴近新一代智能系统“自适应性”“自我进化”的需要。

参考文献:

[1] 肖寒. J2EE平台下代码自动生成技术研究[J]. 电脑知识与技术, 2009, 5(20): 5421-5422, 5434.
XIAO H. Study of code generation technology based on J2EE platform[J]. Computer Knowledge and Technology, 2009, 5(20): 5421-5422, 5434.

[2] CHEN T Q, THIERRY M, JIANG Z H, et al. TVM: an automated end-to-end optimizing compiler for deep learning[C]// The 13th USENIX Symposium on Operating Systems Design and Implementation. [S. l.:s.n.], 2018: 578-594.

[3] 梁青青. 基于关键超参数选择的监督式AutoML性能优化[D]. 贵州: 贵州大学, 2019.
LIANG Q Q. Performance optimization of supervised AutoML based on key super parameter selection[D]. Guizhou: Guizhou University, 2019.

[4] LEMKE C, BUDKA M, GABRYS B. Metalearning: a survey of trends and technologies[J]. Artificial Intelligence Review, 2015, 44(1): 117-130.

[5] 陈森朋, 吴佳, 陈修云. 基于强化学习的超参数优化方法[J]. 小型微型计算机系统, 2020, 41(4): 679-684.
CHEN S P, WU J, CHEN X Y. Hyperparameter optimization method based on reinforcement learning[J]. Journal of Chinese Mini-Micro Computer Systems, 2020, 41(4): 679-684.

[6] 李玉娟. 基于改进粒子群算法的深度学习超参数优化方法[J]. 信息通信, 2020(1): 52-53, 55.
LI Y J. Deep learning hyperparameter optimization method based on improved particle swarm optimization[J]. Information & Communications, 2020(1): 52-53, 55.

[7] 朱汇龙, 刘晓燕, 刘瑶. 基于贝叶斯新型深度学习超参数优化的研究[J]. 数据通信, 2019(2): 35-38, 46.
ZHU H L, LIU X Y, LIU Y. Research on new deep learning super parameter optimization based on Bayes[J]. Shuju Rongkin, 2019(2): 35-38, 46.

[8] 孙晓璇, 张磊, 李健. 目标检测数据集半自动生成技术研究[J]. 计算机系统应用, 2019, 28(10): 8-14.
SUN X X, ZHANG L, LI J. Research on semi-automatic generation technology of object detection datasets[J]. Computer Systems & Applications, 2019, 28(10): 8-14.

[9] 杨懿男, 齐林海, 王红, 等. 基于生成对抗网络的小样本数据生成技术研究[J]. 电力建设, 2019, 40(5): 71-77.
YANG Y N, QI L H, WANG H, et al. Research on generation technology of small sample data based on generative

adversarial network[J]. Electric Power Construction, 2019, 40(5): 71–77.
[10] KURAKIN A, GOODFELLOW I, BENGIO S. Adversarial machine learning at scale[C]// International Conference on Learning Representations. [S.l.:s.n.], 2017.
[11] SU P H, LIU Y H, SONG X. Research on intrusion detection method based on improved smote and XGBoost[C]// The 8th International Conference on Communication and Network Security. [S.l.:s.n.], 2018: 37–41.
[12] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[J]. arXiv preprint, 2015, arXiv: 1412.6572.
[13] CARLINI N, WAGNER D. Towards evaluating the robustness of neural networks[J]. arXiv preprint, 2016, arXiv:1608.04644.
[14] MOOSAVI–DEZFOOLI S M, FAWZI A, FROSSARD P. DeepFool: a simple and accurate method to fool deep neural networks[C]// The IEEE Computer Vision & Pattern Recognition. Piscataway: IEEE Press, 2016: 2574–2582.
[15] SARKAR S, BANSAL A, MAHBUB U, et al. UPSET and ANGRI: breaking high performance image classifiers[J]. arXiv preprint, 2017, arXiv:1707.01159.
[16] CISSEM, ADIY, NEVEROVAN, et al. Houdini: fooling deep structured prediction models[J]. arXiv preprint, 2017, arXiv:1707.05373.
[17] SU J W, VARGAS D V, KOUICHI S. One pixel attack for fooling deep neural networks[J]. arXiv preprint, 2017, arXiv:1710.08864.

作者简介

夏正勋（1979– ），男，星环信息科技（上海）有限公司高级研究员，主要研究方向为大数据、数据库、人工智能、流媒体处理技术等。

杨一帆（1985– ），男，博士，星环信息科技（上海）有限公司产品总监、首席科学家，主要研究方向为统计（统计计算、生存分析、时间序列和生物信息）、机器学习中图计算、强化学习等。

罗圣美（1971– ），男，博士，星环信息科技（上海）有限公司大数据研究院院长，主要研究方向为大数据、并行计算、云存储、人工智能等。

赵大超（1989– ），男，星环信息科技（上海）有限公司产品研发经理，主要研究方向为大数据、人工智能等。

张燕（1985– ），女，星环信息科技（上海）有限公司大数据技术研究员，主要研究方向为大数据、人工智能等。

唐剑飞（1986– ），男，星环信息科技（上海）有限公司大数据技术标准研究员，主要研究方向为大数据、数据库、图计算等。

收稿日期：2020-08-07