

基于神经结构搜索的多目标 认知自动化系统

王柯, 徐鹏



, 陈建明



, 和尼拉杰·库马尔



【摘要】 :当前,基于深度学习的工业信息决策认知自动化是认知计算领域的新热点,其中多目标架构优化是研究领域的难点。

现有算法在面临多目标认知模型问题时,往往需要花费大量时间不断设置不同的搜索偏好参数,以产生新的搜索过程。

本文主要解决多目标神经架构搜索过程中的问题,关键问题是在架构搜索过程中如何适应用户偏好。我们提出了一种新的算法:线性偏好协同进化算法。与原始的用户约束方法和帕累托支配的 NSGA-II 算法相比,我们具有更快的适应时间和更好的适应质量。同时在推理阶段能够以相对较快的速度响应用户的需求。基于大量的对比测试结果,我们的算法在搜索质量的多目标问题上优于传统的认知自动化算法。

索引词 认知自动化、进化算法、多目标、神经结构搜索 (NAS)、帕累托占优。

一、引言

自动化源于人工

认知的模拟人脑的计算机系统的智能。通过人与自然环境的相互作用和不断学习,帮助决策者从不同类型的大量数据中揭示出非凡的见解,从而实现不同程度的感知、记忆、学习等认知活动[1]。大数据时代,数据的规模、种类、速度和复杂性都远超人脑的认知能力。如何有效地实现对大数据的认知,也给传统的认知计算带来了巨大的挑战。

稿件于 2020 年 1 月 28 日收到; 2020 年 6 月 4 日修订; 2020 年 6 月 7 日接受。出版日期 2020 年 6 月 29 日;当前版本的日期为 2021 年 6 月 16 日。这项工作部分得到国家重点研发计划项目 2018YFB1003800 和 2018YFB1003805,广东省自然科学基金项目 2016A030313660 和 2017A030313365,深圳市科技创新基金的部分资助 Grant JCYJ20160608161351559, Grant KQJSCX70726103044992, Grant JCYJ20170811155158682 和 Grant JCYJ20160428092427867 下的项目。(通讯作者: Neeraj Kumar。)

Eric Ke Wang 和 Ship Peng Xu 来自哈尔滨工业大学(深圳)计算机科学系,中国哈尔滨 518055 (电子邮件: wk_hit@hit.edu.cn; 19s051059@stu.hit.edu.cn)。

Chien-Ming Chen 在山东科技大学
中国山东 266510 (电子邮件: chienmingchen@ieee.org)。

Neeraj Kumar 就职于亚洲大学计算机科学与工程系,台湾台中 41354,以及印度帕蒂亚拉 147004 的塔帕尔工程技术学院(电子邮件: neeraj.kumar@thapar.edu)。

数字对象标识符 10.1109/JSYST.2020.3002428

深度学习(DL)作为一种新的机器学习方法,已经成为大数据时代认知计算的绝佳解决方案。DL 通过构建基于表示的多层机器学习模型,训练海量数据并学习有用的特征,以提高识别、分类或预测的准确性[2]。一个优秀的深度学习认知模型,会获得优秀的认知效果。这里最关键的问题是深度神经网络模型的选择。

随着各领域工业应用的快速发展,积累了海量数据。处理和分析数据内容以获取有价值的信息已成为一项关键任务。越来越多的 DL 和神经网络模型已经应用于工业应用。

然而,为工业应用设计和验证一个合适且优秀的基于 DL 的认知自动化模型,往往需要大量的人力,例如选择算法、超参数调整、迭代建模和模型评估。因此,如果有一种方法可以自动找到当前问题的正确解法,就可以有效地节省科学家的人工劳动,解放研究人员的创造力。神经架构搜索(NAS)是一个新的研究方向,也是自动化机器学习的热点之一[3],[4]。通过设计一种性价比高的搜索方法,可以自动得到泛化能力强、硬件友好的神经网络结构,节省了大量的人力成本。其部署和使用更适合产业推广应用。

NAS 旨在解决设计 DL 网络工作严重依赖专家渊博知识的问题,旨在实现具有竞争力的性能,同时自动化神经网络架构设计过程。对于图像识别等一些任务,NAS 已经取得了非常有希望的结果。

NAS 的一个非常重要的分支问题是如何有效地执行多目标 NAS 过程。由于历史原因,关于多目标 NAS 的相关工作很少。然而,许多神经网络的架构往往不仅需要高精度,而且需要低资源消耗。因此,多目标 NAS 是一个需要研究的领域。相比之下,当前的许多算法往往需要用户提前设置准确性和资源消耗之间的权衡。因此,将设置不同的搜索偏好参数以产生新的搜索过程以适应用户的新请求。

由于用户偏好漂移,以往的多目标 NAS 算法往往浪费大量时间适应用户偏好。

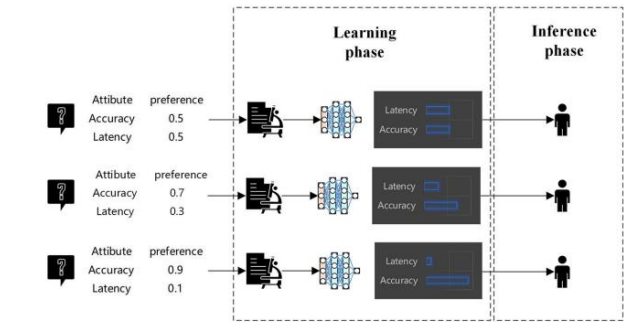


图 1. 给定偏好场景。

如图1所示,在给定偏好的情况下,平台首先获取用户的偏好,然后在用户偏好的条件下进行学习。得到满足用户需求的网络结构后,将网络返回给用户。在这种情况下,对用户偏好的每一次推断都意味着一个耗时的学习过程。在单一推理的过程中,每个用户需要花费更多的资源。而且,面对需要快速部署的用户场景,服务器仍然需要较长的搜索时间,往往无法满足用户快速部署的需求。

A. 贡献

为了解决多目标神经网络架构搜索过程中的关键问题,即在搜索架构时如何自适应地设置用户偏好,我们提出了一种用户偏好延迟方法。

- 1)我们将传统的NAS过程拆分为两个相对独立的阶段:学习阶段和推理阶段。
- 2)在学习阶段,我们使用具有偏好和最优个体的协同进化算法来探索满足用户偏好的解集。
- 3)在推理阶段,我们可以在短时间内使用这个解决方案集响应用户的新偏好。
- 4) 主要目的是满足用户可以更新搜索偏好,系统可以快速适应偏好的问题。在学习阶段 在本文中,我们使用平均自适应质量和平均自适应速度作为指标。本文算法在两个指标上均优于其他算法。

B. 组织

本文的其余部分组织如下。第二节介绍了相关工作。第三部分描述了所提出方案在多目标优化、延迟偏好场景和线性偏好协同进化 (LCNAS) 算法方面的工作。第四节重点介绍了实验结果。最后,第五节总结了本文。

二.相关工作

我们的模型与两个领域相关:多目标优化和 NAS。在接下来的相关工作中,我们主要从两个方面进行介绍。

A. 神经结构搜索

NAS可以分为以下几个子问题:搜索空间的设计、搜索策略的选择、搜索过程的加速。

搜索空间可分为链式搜索空间、多分支搜索空间[5]、[6]、基于基本单元构造的搜索空间[7]和层次表示的搜索空间[8]。NAS的主要方法可以分为三类,使用强化学习和进化学习,以及其他算法方法。在强化学习方面,Zoph等人。[7]使用递归神经网络 (RNN) 作为控制器,将 RNN 生成的序列转换为模型,并通过强化学习学习神经架构。在进化算法方面,Liu等人。[9]和 Real等人。[10]、[11]为进化算法设计了一个搜索空间,并使用遗传算法在这个搜索空间中搜索神经结构。在其他算法中,Hutter等人。[12]使用基于顺序模型的优化作为搜索算法,Liu等人。[13]提出使用可区分的 NAS 来发现硬件感知的高效卷积网络。在 NAS 进程的加速方面,有一些方法被证明是有效的,例如参数共享 [14]、网络态射 [15] 和一次性架构搜索 [16]。

B. 多目标 NAS

目前,多目标NAS常用的算法有两类:一类是约束条件下的搜索算法[17],[18],一类是Pareto边界下的多目标搜索算法[19]-[24]。

在用户约束算法中,用户往往需要预先设置一系列约束或偏好,然后将神经结构搜索过程构建为特定约束下的一次性搜索过程。它导致两个主要问题:1)搜索过程只能找到满足单一约束或用户偏好的神经网络架构。当用户的限制或偏好发生变化时,需要研究搜索过程;和 2) 约束或偏好的选择通常是问题相关的,因此没有关于问题特征的信息很难进行搜索过程。

一方面,Pareto 占优策略算法 (如 NSGA-II [27])适用于低维多目标优化问题。然而,对于高维多目标优化问题,使用帕累托优势往往会导致弱选择压力。此外,在高维空间中,使用Pareto算法的计算复杂度也比较高。

为了在搜索时间上获得相对更多的满足用户偏好的搜索结果,同时降低搜索计算复杂度,我们提出了LCNAS算法。

C. 延迟线性偏好情景

Yang 的工作 [30] 引入了延迟线性偏好场景,用于构建多目标强化学习算法。我们将这个场景简化为两个阶段,用它来

重建NAS流程,介绍下这个场景的初衷。

偏好方程如下: $t: R^m \rightarrow R$ 表示 m , 它将目标向量映射到单个标量。

给定一个可行解 x 及其目标向量 $y^\wedge = F(x) = (y_1, \dots, y_m)$, 该策略的效用可以定义为 $t(y^\wedge)$ 。
对于可行解 x , 该可行解的实际效用是 $t\omega(x) = \omega x$, 其中偏好权重为 $\omega \in \Omega$, $\omega = (\omega_1, \dots, \omega_n)$, $\omega = 1$ 。在这种特殊情况下, 偏好解只能来自帕累托边界的凸覆盖集 (CCS)。

在多目标优化问题中, 给定其 Pareto 边界 P^* , 这个 CCS 可以定义为

$$CCS := \{x^\wedge \in P \mid \exists \omega \in \Omega \text{ st } \omega T F(x^\wedge) \geq \omega T F(x^\wedge), \forall x^\wedge \in P\} \quad (1)$$

延迟线性偏好的另一个关键特征是延迟。在原文中, Yang 将延迟线性偏好场景分为三个阶段: 学习阶段、分析阶段和执行阶段。被延迟意味着线性偏好最初是未指定的, 直到场景的后期阶段才会给出特定的线性偏好。

D. 使用自适应权重的基于分解的多目标进化算法 (MOEA)

为了在学习阶段有效地研究 CCS, 我们使用偏好协同进化策略。事实上, 用户偏好协同进化的想法源于使用自适应权重的基于分解的 MOEA。偏好权重可以看作是用于激励算法获得更好结果的权重, 也就是说它可以引导进化算法沿着给定的方向进行优化。

有多种方法可以适应权重。张等。 [25] 提议将动态权重设计方法纳入 MOEA/D (表示为 DMOEA/D), 以使 MOEA/D 能够在具有不同几何形状的问题上表现良好。王等。 [26] 提出了一种使用权重的偏好启发协同进化算法, 其中权重在搜索过程中与候选解共同进化。

然而, 大多数使用自适应权重算法的基于分解的 MOEA 都试图使用协同进化算法来获得 Pareto 边界, 而忽略了基于权重的方法在解决凸问题方面天然具有优势的事实。

因此, 我们在 NAS 任务中应用基于权重的算法进行近似。

三. 方法论

在本节中, 我们将说明如何将 NAS 形成延迟线性偏好场景问题, 并介绍基于给定偏好的场景与基于延迟偏好的场景之间的区别。在制定问题之后, 我们提出了一种称为线性偏好协同进化算法的新算法, 用于在学习阶段有效地搜索 CCS。

A. 多目标优化

首先, 我们将给出多目标优化问题的一般描述。对于一个有 n 个决策变量和 m 个目标变量的多目标优化问题, 我们可以表示如下:

$$\begin{aligned} \text{最大化} \quad & y = F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad g_i(x) \leq 0, i = \\ \text{英石} \quad & 1, 2, \dots, q \quad h_j(x) = 0, j = 1, 2, \dots, p, x = (x_1, \dots, x_n) \in X \subset R^n \end{aligned} \quad (2)$$

Y 是 m 维目标空间。目标函数 $F(x)$ 定义了 m 个从决策空间到目标空间的映射函数; $g_i(x) \leq 0 (i = 1, 2, \dots, q)$ 定义 q 个不等式约束; $h_j(x) = 0 (j = 1, 2, \dots, p)$ 定义 p 等式约束。

B. 延迟偏好情景

通常, 用户根据自己的用户需求上传到平台, 平台处理用户的需求, 然后进行一次性搜索, 找到合适的架构返回给用户。不同用户的需求往往是不同的, 用户的需求会随着时间的推移而改变。

它使用数学语言对 ML 进行了如下描述:

$$m \in ML \Rightarrow \exists \omega \in \Omega \text{ st } \forall m \in M, \omega F(m) \geq \omega F(m) \quad (3)$$

仅当该偏好方程下至少存在一个线性偏好 ω 时, M 才包含在 ML 中; 没有其他个体 m 可以产生更高的实用性。

在学习阶段, 搜索的计算资源相对充足, 但线性偏好此时未知。对于多目标进化算法, 我们关注是否可以用 ML 近似 CCS。

在这个阶段, 多目标进化学习的目标是获得一个 ML 来近似 CCS。对于任何相关的偏好向量 ω , 在 ML 中总有一个是对 ω 最好的结果。一般来说, 学习阶段是为处理未来的各种偏好而准备的。

在推理阶段, 将给出一个线性偏好方程 $f_\omega(\cdot) = \omega \cdot$ 。将使用更少的时间来给出最佳个体 m_ω 作为对已经学习的策略集 ML 的响应。

我们将这种为特定偏好选择最优策略的过程称为偏好适应。因为需要学习的 CCS 可能非常大, 所以通过之前学习的 ML 来有效地选择当前偏好下的最佳个体并不是一件容易的事。

在基于图2的延迟偏好的情况下, 平台只需要记录一次学习过程中的最优个体集合 ML。在推理阶段, 可以用非常有限的资源消耗来适应每个用户。

也就是说, 在这种场景下, 平台可以高效快速地推断出当前用户偏好下的最优解。因此, 基于延迟偏好场景, 虽然在学习过程中可能会带来更多的资源消耗

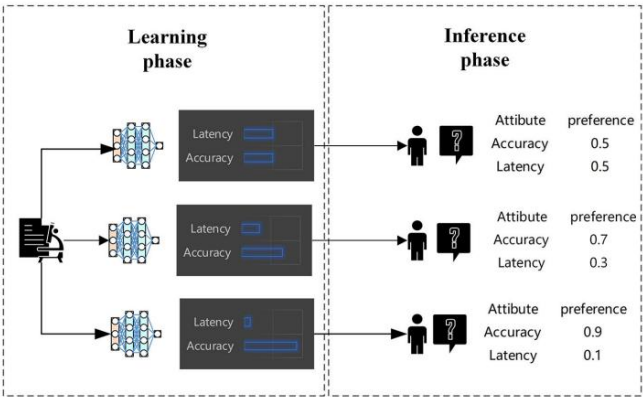


图 2. 延迟偏好场景。

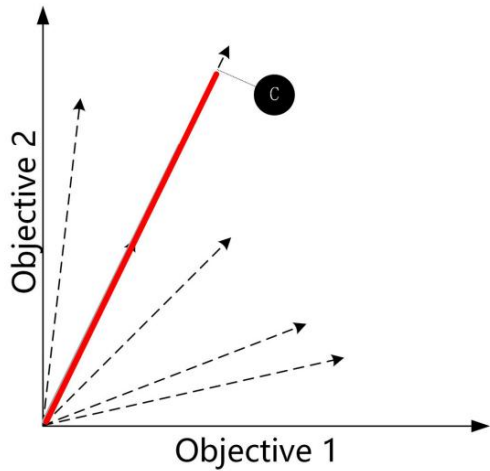


图 3. 最大效用个体函数的描述。

阶段,如果考虑到大量的用户和这些用户不断变化的需求,延迟偏好的场景对神经网络搜索有很大的意义。它可以节省更多的计算资源,更快地响应用户的部署需求,更好地处理不断变化的用户偏好。

C. LCNAS 算法

在学习阶段,一个重要的问题是如何有效地使用进化算法在给定的一组偏好上近似 CCS。为了解决这个问题,我们提出了一种线性偏好的层次排序算法。通过该算法,我们可以有效地选择在线性偏好下占优势的个体。另一个重要问题是多目标进化算法往往只在给定偏好的方向上进行优化,这导致解决方案无法有效覆盖整个 CCS。为了解决这个问题,我们采用了偏好共同进化的策略。在此策略中,我们在每次迭代期间生成一组新的偏好。然后,我们将选择能够使所选个体实现最大效用的偏好,如图 3 所示。

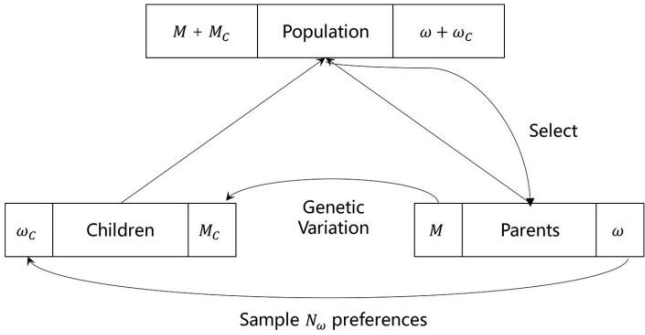


图 4. LCNAS 的实现框架。

算法 1: LCNAS 算法。

```
输入:偏好抽样分布Dω,小批量偏好NΩ,每一代个体数量 N,
初始种群M0 PMF = ∅ M = M0 FM =
objectiveFuction (M)

PMF* = updatePMF(PMF*, FM)
对每个t = [ 1, 2, ..., T]做

    MC =生成新流行音乐 (M)
    FMC = objectiveFuction (MC )
    M = M ∪ MC
    FM = FM ∪ FMC
    采样NΩ偏好,其中 ω Dω为ΩC Ω=Ω ∪ ΩC

    (FM,M,Ω) =共同进化(FM,M,Ω)
    PMF* = updatePMF (PMF*,M,FM, Ω) end
for

输出: PMF*
```

具体来说,我们在图4所示的框架内实现了LCNAS算法。

M 和 Ω 是固定大小的 N 和NΩ向量,代表候选个体的数量和偏好。在每一代 t,将遗传变异算子应用于父母 M,产生 N 个后代MC。同时生成NΩ个新的权重向量ωC。M 和MC,Ω 和ΩC,将分别加入。然后使用线性偏好算子对种群进行排序,选择最好的 N 个个体作为新的候选种群 M。选择NΩ偏好作为新的偏好向量 Ω,这样,我们将使用解决方案。

LCNAS 的伪代码如算法 1 所示。LCNAS 的核心部分在于函数 coEvolve,接下来将对其进行详细说明。

1)Coevolve算法:函数coEvolve评估候选个体的表现和偏好,然后分别选择新的父种群M和Ω。父种群 M 将使用线性优先前排序进行排序。当将选定的候选解决方案排名为最佳时,将选择偏好。

函数 coEvolve 的伪代码如下。

```
算法 2: coEvolve 算法。
输入:偏好向量  $\Omega$ ,候选解  $M$  及其目标值  $FM$ 
 $F = \text{linear-dominate-front-sort}(M, \Omega)$ 

 $Mc+1 = \emptyset$  且  $j=1$  而  $|Mt+1| + |Fd| \leq N$  do
   $Mt+1 = Mt+1 \cup F4$ 
   $i=i+1$  end while  $Mt+1 = Mt+1 \cup F[1 : (N - [Mt+1])]$ 

 $M = Mt+1$ 
for  $m \in M$  do
   $\omega = \text{selectP}(m, \omega)$ 
end for
Output:  $\Omega$ ,  $M$ ,  $FM$ 
```

1) 第 1 行应用函数linear-dominate-front-sort使用线性支配运算符对个体进行排序。 linear-prefer-front-sort 的伪代码在算法 3 中表示。我们将linear-dominate算子定义为linear-dominate(m, w, Ω),其中个体 $m, w \in M, \Omega$ 是偏好向量

线性主导 (m, w, Ω)

= 是的, $\omega_{TF}(m) \geq \omega_{TF}(w), \forall \omega \in \Omega$.
假的,否则

2) 第 2 行应用函数selectP从 W 中为每个幸存的候选解决方案选择最合适的偏好向量。我们将最大个体效用函数定义为selectP(m, ω),其中个体 $m \in M, \omega$ 是对偏好分布 $D\omega$ 的偏好:

= max $\omega_{TF}(m) \omega \in \omega, \text{选择P}(M, \omega)$ (4)

四.实验和结果

A. 实验设置

本节实验的设置是为了探索perform LCNAS 在给定数据集 CIFAR-10 下的 mance。 在我们的工作中,我们使用 NASBench 来评估性能数据。 NASBench 是一个表格数据集,它将卷积神经网络架构映射到它们在 CIFAR-10 上经过训练和评估的性能。

对于这两种算法,我们设置初始种群数 $N = 100$ 和 $T = 10$ 次的遗传过程,其中 $pM = 0.8$, $qM = 0.1$, $pC = 0.2$, $qC = 0.3$ 。对于preference adaptation的环境,我们设置preference number为 $N\omega = 100$ 。我们对两种算法有效性的测试是测试模型的(acc(m), trainingTime(m), and param(m))通过固定的训练轮次、固定的种群数量和遗传过程的数量得到。(acc(m), trainingTime(m), param(m)分别对应神经网络的三个指标模型:

```
算法 3:线性支配前排序算法。
输入:  $M$ ,及其目标值 $FM$ ,偏好向量  $\Omega$ 
对于 $m \in M$  do  $Sm = \emptyset$ 
  对于 $w \in M$  do
    如果linear-dominate( $m, w$ )那么
       $Sm = Sm \cup \{w\}$ 
    else if linear-dominate( $w, m$ ) then
       $nm = nm + 1$  end if
  end if
  if  $nm = 0$  then
     $mrnk = 1$ 
  end if
   $F1 = F1 \cup \{m\}$ 
end for
j=1而 $Fi = \emptyset$  do 则结束

 $W = \emptyset$ 
对于 $m \in M$  do
  对于 $w \in M$ 做
     $nm = nm + 1$  如果
     $nm = 0$ 则 $i = i + 1$ 
     $w$ 坦克
     $W = W \cup \{w\}$ 
  end if
end for
for end while end
for end for

输出:  $\Omega, M, FM$ 
```

精度、训练模型的收敛时间和参数数量[31]。对于普通的遗传算法,我们采取三次取最优组的方法。

B. 评估指标

在我们的工作中,我们引入了两个指标来评估 LCNAS 在多目标 NAS 下的定量性能。 第一个指标是适配质量,用于评价同一用户偏好下不同算法的平均效用。第二个指标是适应速度,即学习阶段和推理阶段所花费的平均时间,对应于一组用户偏好。

1) 自适应效用 (AU):对于来自 n 个用户偏好的偏好集 $\Omega_u = (\omega_1, \dots, \omega_n)$,我们使用多目标进化算法 A 来学习个体集合,它近似于 Ω_u 下的 CCS。在推理阶段,我们可以得到一组响应个体 $M_u = m_1, \dots, m_n$ 然后,我们可以将 AU 定义为

$$AU(A) = \frac{1}{n} \sum_{i=1}^n F(\omega_i, A)$$
 (5)

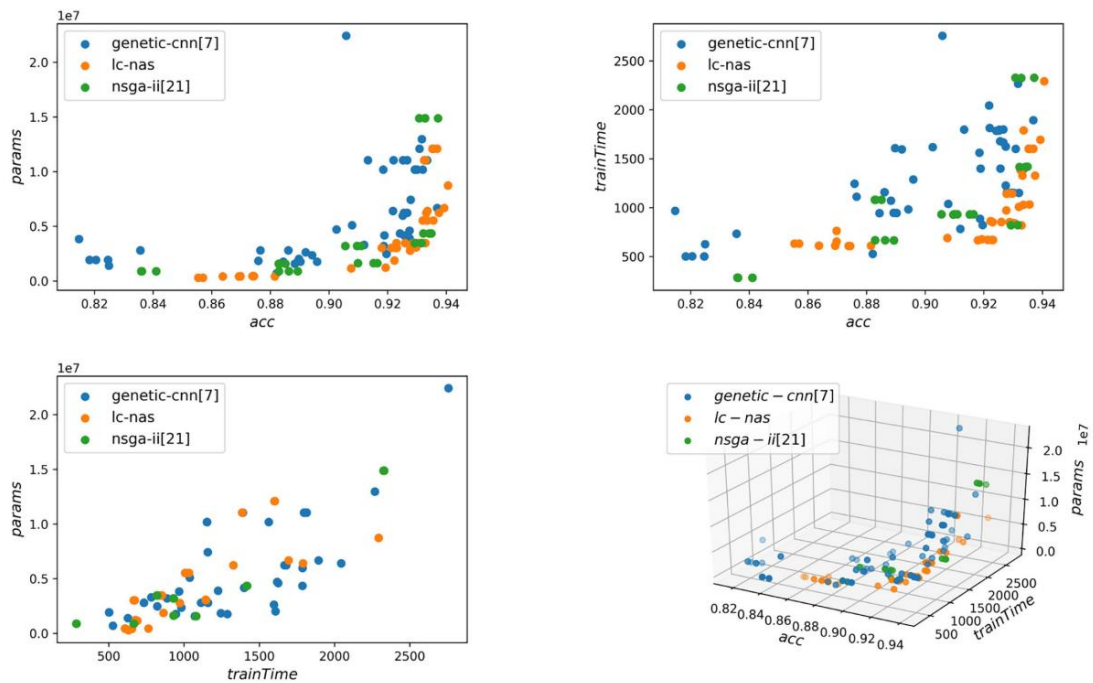


图 5. 解决方案集可视化。

其中, F 为目标函数, 表示该结构在NAS中的准确率和资源消耗指标。

2) 自适应时间: 对于来自 n 个用户偏好的一组 $Q_u = (\omega_1, \dots, \omega_n)$, 我们使用多目标进化算法 A 来学习这组偏好, 并使用 t_L 表示学习阶段所用的时间。在推理阶段, 我们可以使用响应每个偏好的时间成本作为 $t_1 \dots t_n$ 。那么, 我们可以定义自适应时间为

$$AT(A) = t_L + \sum_{i=1}^n t_i \quad (6)$$

3) 覆盖率: 覆盖率是衡量多目标算法找到所有潜在最优解的能力。对于多目标进化算法 A , 我们用 M 表示算法得到的最优种群集合, 用 M_{target} 表示期望的目标种群集合。我们定义覆盖率可以用 F1 分数的形式来定义: $\text{precision} \cdot \text{recall}$ $CRA(M, M_{target}) = 2 \cdot \text{precision} + \text{recall}$

$$\text{precision} = \frac{|M \cap M_{target}|}{|M|}, \text{recall} = \frac{|M \cap M_{target}|}{|M_{target}|} \quad (7)$$

其中 $\text{precision} = |M \cap M_{target}| / |M|$, 表示最优解在解集中的比例, $\text{recall} = |M \cap M_{target}| / |M_{target}|$, 表示最优解的比例得到整个最优解。

C. 实验结果与实验分析

首先, 在图 5 中, 我们可视化了经过 100 代迭代后使用遗传-CNN [4]、LCNAS 和 NSGA-II [27] 算法最终获得的解集。它是成对指标的比较, 包括精度和参数对、精度对和训练时间。

表一
公制

N_u	AU	
	genetic-cnn[4]	lc-nas
1	0.190	0.191
2	0.274	0.275
4	0.353	0.355
8	0.290	0.291
16	0.313	0.314
32	0.341	0.342
64	0.307	0.308

从直观的角度来看, 我们可以推测使用 LCNAS 获得的解集质量更高, 意味着使用 LCNAS 获得的解集更准确, 参数和训练时间更少。为了对解集进行定量评价, 引入 AU 的概念来评价算法得到的解集的好坏。

但是, 如果我们想使用 AU 来比较这两种 NAS 算法, 我们必须进行归一化操作。原因是上一节提到的 NAS 的三个参数, 即准确率、训练时间和参数数量, 往往有不同的数量大小。这里, 我们选取了两种算法的最后一代的数据进行归一化, 归一化后的结果如图 6 所示。

在使用归一化操作后, 我们比较了两种算法的 AU 值, 如表一所示。从表中可以看出, 随着用户偏好数量的增加, 两种算法的平均效用值呈现出先增加, 然后减少, 然后稳定。然后, 在每个用户偏好数下, LCNAS 的平均效用高于遗传-CNN 算法。因此, 给定一组偏好, LCNAS 可以实现更大的平均效用。

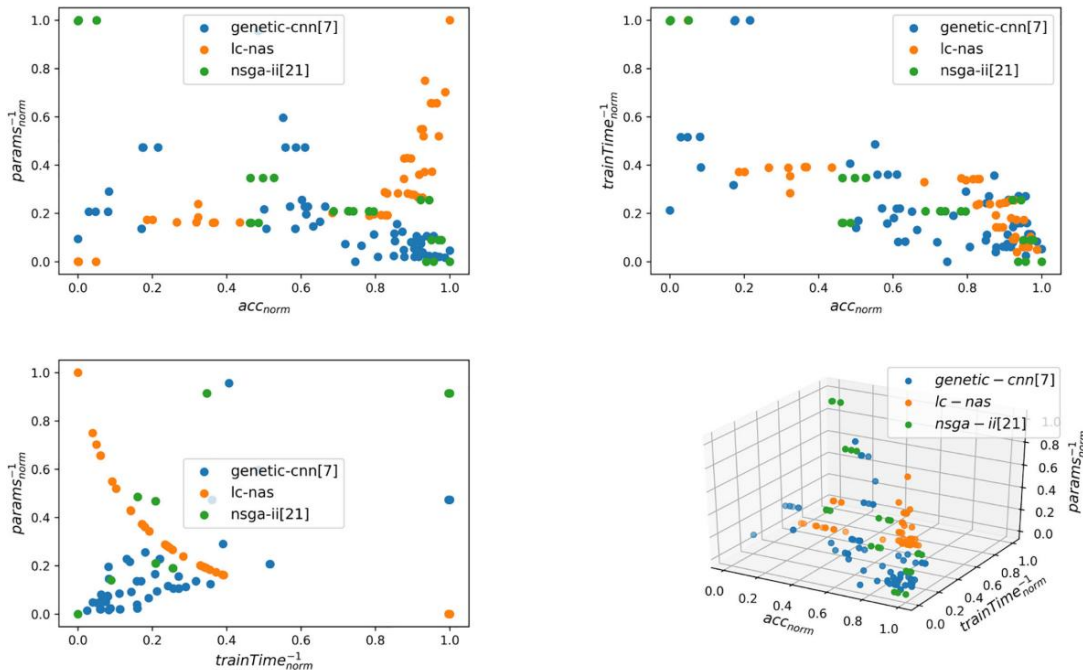


图 6. 归一化结果。

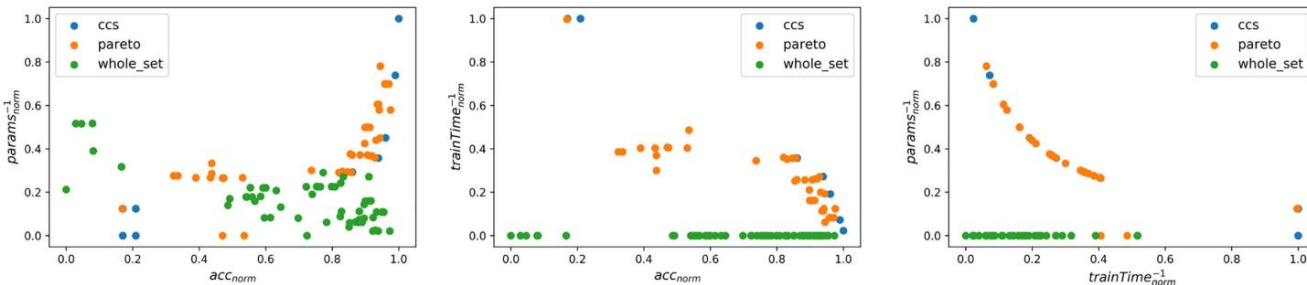


图 7. Pareto 和 CCS 可视化。

表二
与非线性支配算法的度量比较

	Precision		recall		Coverage Ratio	
	CCS	Pareto Front	CCS	Pareto Front	CCS	Pareto Front
genetic-cnn[4]	0.270	0.230	0.338	0.404	0.300	0.293
lc-nas	0.520	0.510	0.863	0.930	0.649	0.659
nsga-ii[27]	0.530	0.470	0.663	0.825	0.589	0.599

此外,为了比较两种算法在同一维度的搜索能力,我们引入了覆盖率的概念.首先,我们将两种算法得到的上一代解集进行合并归一化,得到解集.然后,我们在这个解集中找到对应的Pareto解集和CCS解集.我们将合并集的pareto 解集和 CCS 解集可视化,如图 7 所示。

我们使用覆盖率指标对两种算法得到的解集进行评估,结果如表2所示。

其中,第一行显示了两种算法对CCS的逼近效果.我们可以看到 LCNAS 在召回率和覆盖率方面表现更好.第二个显示

两种算法对Pareto前沿的逼近效果。LCNAS 算法在一定程度上更好地逼近 Pareto 边界,比 CNN 算法获得更高的召回率和准确率。

D. 结果和与其他方法的比较

为了进一步评估我们提出的算法的性能,我们在 CIFAR-10 上进行了另一个实验.在以下方法和手动模型之间进行了性能比较,其中主要关注参数的数量和准确性:

- 1)架构模型LC-NET-V1,由LC获得 NASBENCH101搜索空间中的NAS方法;
- 2) MobileNet V1.V2.[28].[29],它们是手动设计的架构,旨在减少参数同时保持高预测性能;
- 3) NASNet [7],它是通过强化学习的 NAS,之前在 CIFAR-10;

表三
模型比较

MODEL	PARAMS	ACC (%)	PARAMS/ACC
MOBILENET	834K	95.0	8.78
MOBILENET V2	850K	95.4	8.91
NASNET	926K	95.3	9.72
RANDOM SEARCH	1200K	94.7	12.67
DPP-NET	1.0M	95.3	10.5
LC-NAS	873K	94.1	9.27

4)DPP-net[19],是Dong等人提出的一种多目标NAS方法; 5)NAS的随机搜索基线。

表 III 显示了性能比较的总结

在上述方法和手动模型之间进行实验。

如表 III 所示,LC-NET-V1 以相对较少的参数实现了相当高的精度。此外,LC-NET-V1 获得的参数量最少,与其他 NAS 方法相比,接近于为移动性能手动设计的 MobileNet V1 和 MobileNet V2。

在 NAS BENCH101 相对有限的搜索空间下,通过 LC-NAS 方法搜索了这个相对较好的模型。它在准确性方面与其他消耗大量GPU模型的搜索方法处于同一水平,在参数方面优于其他方法。

五.结论

在本文中,我们针对工业信息学的原始多目标认知自动化提出了一种偏好协同进化算法和一个新场景,将**认知自动化过程**分为学习阶段和推理阶段两个步骤。在**学习阶段**,采用偏好协同进化算法探索满足用户偏好的解集。在**推理阶段**,算法可以使用该解决方案集在**短时间内响应用户的新偏好**。实验结果表明我们的算法优于其他算法。与原有的用户约束方法和帕累托支配的NSGA-II算法相比,具有更快的适应时间和更好的适应质量。

参考

[1] A. Fasth-Berglund 和 J. Stahre, “可重构和可持续装配系统的认知自动化策略”, Assembly Autom., 卷. 33, 没有. 3, 第 294–303 页, 2013 年。

[2] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu 和 FE Alsaadi, “深度神经网络架构及其应用的调查”, Neurocomputing, 卷. 234, 第 11–26 页, 2017 年。

[3] Y. He 等人, “AMC:用于移动设备上的模型压缩和加速的 AutoML”, Proc. 欧元。会议。电脑。Vis., 2018 年, 第 815–832 页。

[4] T. Elsken, JH Metzen 和 F. Hutter, “神经架构搜索:一项调查”, J. Mach. 学习。水库, 卷. 20, 第 1–21 页, 2019 年。

[5] K. He, X. Zhang, S. Ren 和 J. Sun, “用于图像识别的深度残差学习”, Proc. IEEE 会议。电脑。可见。模式识别, 2016 年, 第 770–778 页。

[6] B. Li, K. Xu, X. Cui, Y. Wang, X. Ai 和 Y. Wang, “基于多尺度 DenseNet 的窃电检测”, 智能计算理论与应用。Cham, 瑞士:施普林格出版社, 2018 年。

[7] B. Zoph, V. Vasudevan, J. Shlens 和 QV Le, “学习用于可扩展图像识别的可迁移架构”, Proc. IEEE 会议。电脑。可见。模式识别, 2018 年, 第 8697–8710 页。

[8] C. Liu 等人, “Auto-deepLab:用于语义图像分割的分层神经架构搜索”, Proc. IEEE 会议。电脑。可见。模式识别, 2019 年, 第 82–92 页。

[9] H. Liu, K. Simonyan, O. Vinyals, C. Fernando 和 K. Kavukcuoglu, “高效架构搜索的层次表示”, Proc. 诠释。会议。学习。陈述, 2018 年, 第 13–25 页。

[10] E. Real, A. Aggarwal, Y. Huang 和 QV Le, “图像分类器架构搜索的正则化演化”, Proc. 第 33 届 AAAI 会议。神器。Intell., 2019, 第 4780–4789 页。

[11] Esteban Real 等人, “图像分类器的大规模演化”, Proc. 第 34 国际会议。马赫。学习, 2017 年, 第 2902–2911 页。

[12] F. Hutter, HH Hoos 和 K. Leyton-Brown, “通用算法配置的基于顺序模型的优化”, Proc. 诠释。会议。学习。智能。优化, 2011 年, 第 507–523 页。

[13] H. Liu, K. Simonyan 和 Y. Yang, “DARTS:可微架构搜索”, Proc. 诠释。会议。学习。陈述, 2019 年, 第 106–118 页。

[14] H. Pham, MY Guan, B. Zoph, Q. Le 和 J. Dean, “通过参数共享进行高效的神经架构搜索”, Proc. 第 35 国际会议。马赫。学习, 2018 年, 第 4095–4104 页。

[15] C. Liu 等人, “渐进式神经架构搜索”, Proc. 欧元。会议。电脑。可见, 2018 年, 第 19–35 页。

[16] M. Cho, M. Soltani 和 C. Hegde, “通过压缩感知进行一次性神经架构搜索”, 2019 年, arXiv: 1906.02869。

[17] M. Tan 等人, “MnasNet:面向移动设备的平台感知神经架构搜索”, Proc. IEEE 会议。电脑。可见。模式识别, 2019 年, 第 2815–2823 页。

[18] L. Tan, B. Brotherton 和 T. Sherwood, “用于入侵检测和预防的位分割字符串匹配引擎”, ACM Trans. 建筑师。代码优化, 卷. 3, 第 3–34 页, 2006 年。

[19] J.-D. 董, A.-C. Cheng, D.-C. Juan, W. Wei 和 M. Sun, “DPP-Net:帕累托最优神经架构的设备感知渐进式搜索”, Proc. 欧元。会议。电脑。可见, 2018 年, 第 517–531 页。

[20] N. Kumar, N. Chilamkurti 和 SC Misra, “物联网贝叶斯联盟博弈:基于环境智能的评估”, IEEE Commun. 杂志, 卷. 53, 没有. 1, 第 48–55 页, 2015 年 1 月。

[21] N. Kumar, AV Vasilakos 和 JJPC Rodrigues, “基于多租户云的 DC 纳米电网,用于智能城市中的自持智能建筑”, IEEE 通讯。杂志, 卷. 55, 没有. 3, 第 14–21 页, 2017 年 3 月。

[22] J. Lian 等人, “通过基于局部变化的夸大的生成对抗网络进行基于深度学习的小表面缺陷检测”, IEEE Trans. Ind. Inform., 卷. 16, 没有. 2, 第 1343–1351 页, 2020 年 2 月。

[23] A. Makkar 和 N. Kumar, “基于用户行为分析的网页排名智能能源管理:基于学习自动机的解决方案”, 维持。计算机。通知。系统, 卷. 20, 第 174–191 页, 2018 年。

[24] T. Elsken, JH Metzen 和 F. Hutter, “通过拉马克进化进行高效的多目标神经架构搜索”, Proc. 诠释。会议。学习。陈述, 2019 年, 第 1442–1464 页。

[25] Q. Zhang 和 H. Li, “MOEA/D:基于分解的多目标进化算法”, IEEE Trans. 进化。计算机, 卷. 11, 没有. 6, 第 712–731 页, 2007 年 12 月。

[26] R. Wang, RC Purshouse 和 PJ Fleming, “使用权重向量的受偏好启发的共同进化算法”, Eur. J. 歌剧。水库, 卷. 243, 没有. 2, 第 423–441 页, 2015 年。

[27] K. Deb, A. Pratap, S. Agarwal 和 T. Meyarivan, “一种快速的精英多目标遗传算法:NSGA-II”, IEEE Trans. 进化。计算机, 卷. 6, 没有. 2, 第 182–197 页, 2002 年 4 月。

[28] AG Howard 等人, “MobileNets:用于移动视觉应用的高效卷积神经网络”, 2017 年, arXiv: 1704.04861。

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov 和 L. Chen, “Mo bileNetV2:倒置残差和线性瓶颈”, Proc. IEEE 会议。电脑。可见。模式识别, 2018 年, 第 4510–4520 页。

[30] R. Yang 等人, “一种用于多目标强化学习和策略适应的通用算法”, Proc. 会议。进阶神经信息。过程。系统, 2019 年, 第 14636–14647 页。

[31] B. Wu 等人, “Shift:空间卷积的零 FLOP, 零参数替代方案”, Proc. IEEE 会议。电脑。可见。模式识别, 2018 年, 第 9127–9135 页。