

重访Q-NAS:探索进化适应性以提高效率

丹妮拉·斯瓦克曼
PUC-Rio, IBM Research 里约
热内卢, 巴西
daniela.szw@ibm.com

丹尼尔奇维塔雷塞
IBM 研究院
巴西里约热内卢
sallesd@br.ibm.com

Marley Vellasco 电气
工程系 PUC-Rio

巴西里约热内卢
marley@ele.puc-rio.br

摘要 在过去十年中,科学界见证了深度神经网络在各种任务中的成功。然而,这些新结构的设计仍然需要专业知识和大量时间。在这种情况下,自动化设计此类复杂网络的想法激发了许多研究人员的灵感。已经提出了新的算法来解决神经架构搜索问题,但计算成本是一个主要缺点。Q-NAS (Quantum-inspired Neural Architecture Search)是最近提出的一种进化算法,其思想是提高效率,但在搜索中引入了较小的人为偏差。这项工作扩展了 Q-NAS 的分析,重点关注进化过程中的适应性行为。我们想验证是否在进化的早期就获得了好的个体,所以我们可以应用早期停止机制。我们试验了一种简单的提前停止技术,结果表明在大多数情况下进化时间减少了 45% 以上。

能够平衡效率和通用性的算法代表了 NAS 当前最先进的技术进步。

考虑到这种平衡的重要性,[14] 中的作者提出了 Q-NAS (受量子启发的神经架构搜索),它将受量子启发的进化算法 (QIEA)应用于 NAS 问题。QIEA 是一类进化方法,它使用量子计算的思想,例如状态叠加 [15]、[16]。与相关算法相比,实证研究证实,对于许多优化问题,QIEA 可以通过更少的评估找到更好的结果 [17]、[18]。Q-NAS 的作者声称这是朝着 NAS 效率迈出的充满希望的新一步,人为偏差较小 [14]。他们在 CIFAR-10 数据集集中报告了一个少于 20 层和 89% 准确率的最终网络,使用 20 个 K80 GPU 大约两天。

这个新特性可以提高效率,使 Q-NAS 与其他算法相比非常有竞争力。

索引词 神经结构搜索,进化算法
数学、深度学习、量子启发算法

一、引言

AlexNet [1]、VGGNet [2] 和 ResNet [3] 等架构的出色性能证实了卷积神经网络 (CNN) 最近在图像应用方面的成功。这些网络是手工设计的,它们的整体结构仍然可以与第一个实用的 CNN LeNet 模型 [4] 相媲美。

神经架构搜索 (NAS) 是自动化神经网络设计的过程。NAS 是机器学习社区越来越感兴趣的领域,因为深度系统的设计需要专业知识和大量时间 [5]。为了解决 NAS 问题,研究人员提出了基于不同技术的算法,包括强化学习 (RL) [6]–[8] 和进化算法 (EA) [9]、[10]。与最先进的手工架构相比,几种方法呈现出有竞争力的结果。

然而,其中许多需要大量的计算资源 [6]、[9]–[11]。

最近,出现了考虑计算成本问题来解决 NAS 的新解决方案。新想法包括使用网络转换 [12]、提前停止技术 [8]、块搜索 [8]、[13] 等。在大多数情况下,提高效率的权衡是增加人为偏见以减少搜索空间。因此,新

在这项工作中,我们扩展了[14]中提出的分析,重点分析了进化过程中的适应度函数。这个想法是为了验证 Q-NAS 是否可以从提前停止机制中受益。如果在进化过程中相对较早地找到最终网络,Q-NAS 将通过一种可以识别停滞并更快停止算法的机制来提高效率。此外,我们试验了精英主义选择,它比 [14] 中使用的稳态更保守。我们想检查更多样化的人口是否有助于更快地找到更好的结构。

本文有四个附加部分,组织如下。第二节概述了 NAS 的相关工作; Q-NAS 算法在第 III 节中描述;第四节详细说明实验并讨论结果;最后,第五节给出了结论。

二、相关工作

神经网络的自动设计之前已经被研究过,正如已经发表的关于神经进化的大量工作所证实的那样。这些方法应用 EA 来改进网络权重及其结构 [19]、[20]。然而,以前的神经进化方法仅限于较小的网络,无法扩展到新的深层结构。

最近的 NAS 研究完全集中在深度复杂网络的自动工程上。

Zoph 和 Le [6] 几年前提出了一项关于 NAS 的重要工作,展示了手工设计架构的竞争结果。作者应用了强化

学习技术来搜索性能良好的深度模型。他们使用递归神经网络作为 RL 系统中的控制器来生成可变长度字符串来编码网络结构。在图像分类的情况下,生成的结构在完整的 CIFAR-10 [21] 上进行训练,准确性是强化学习系统中的奖励。尽管他们的结果具有相关性并刺激了 NAS 研究,但他们使用了 800 个 GPU 超过 3 周才实现了这些结果。

进化技术也同样应用于 NAS,进化仅发生在结构空间中,权重通过常规的基于梯度的训练进行调整 [5]。

在[9]中提出的进化方法中,作者使用了一种一次性技术,在搜索结束时产生一个完全训练的网络。该方法从只有一层的网络开始,突变算子允许它们增长。

他们使用 250 个 GPU 耗时 256 小时来搜索他们最好的网络,声称他们的搜索空间很大。
NAS 方法的高计算成本激发了科学家们创造旨在提高效率的新算法。

伊斯特拉特等人。 [22] 引入了一种准确度预测器,无需训练即可估计网络的分类性能。他们取得了可喜的成果,尤其是在效率方面。然而,他们的方法依赖于已知数据集的外部数据库和按难度对它们进行排名的指标。

提高效率的一个反复出现的想法是限制搜索空间。例如,优化块或图案而不是完整的结构 [10]、[11]、[13]。通过以预定义的方式堆叠候选块形成网络,确定所谓的元架构。即使存在强烈的人为偏见来限制搜索空间 [5],计算资源仍然具有表现力:Liu 等人。 [13] 使用 200 个 GPU 1.5 天,在 CIFAR-10 上取得了最好的结果;佐夫等人。 [11] 在 500 个 GPU 上使用了 4 天;和 Real 等人。 [10] 使用 450 个 GPU 进行了 7 天的实验。

三、 Q-NAS算法

A. 受量子启发的进化算法 受量子启发的计算利用量子物

理学的原理来创建可以在经典计算机上执行的算法 [23]、[24]。 QIEA 应用这些原则来解决优化问题。与其他进化算法类似,QIEA 的特征是代表问题可能解决方案的个体群体。规范的 EA 可以概括如下: (1) 生成初始种群; (2) 对个人进行评价和排名; (3) 利用重组或突变改变种群; (4) 为下一个种群选择个体; (5) 重复[2, 3, 4]直到达到停止条件。

QIEA 和其他 EA 之间的一个关键区别是量子总体,它代表搜索空间中状态 (可能的解决方案)的叠加 [16]。请注意,由于一个量子个体编码了许多解,因此只有当它坍缩到一个单一状态时才能对其进行评估,从而生成一个经典个体。创建经典种群的观察过程取决于所选的量子表示。

量子个体可以用一串例如,q 位。单个 Q 位 $[\alpha \beta]$ 的状态 $|\psi\rangle$ 定义为 [15]、[16]:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \tag{1}$$

其中 α 和 β 是复数。Q 位以概率 $|\alpha|^2$ 坍缩到状态 “0”并以概率 $|\beta|^2$ 进入状态 “1” 2. 当进化开始时,Q比特个体以相同的概率代表搜索空间中所有量子算子可以修改每个 q 位的概率,因此它可以逐渐收敛到一个单一状态,该状态应该代表它正在编码的变量的最优解。

已经引入了具有其他量子表示的 QIEA,例如实变量的数值表示 [17]、[18],以及二进制和数值的组合,用于具有分类变量和实变量的任务 [20]、[25]。

B. Q-NAS 步骤

Q-NAS 是一种受量子启发的进化算法,专注于深度神经架构搜索问题,目前在图像分类任务的背景下构建 [14]。我们首先概述 Q-NAS 算法;量子人口和候选网络表示的详细信息将在本节后面进行审查。

图 1 显示了 Q-NAS 的总结步骤。该算法从初始化量子总体 Q(t) (第 2 行)开始,包括为每个量子个体分配初始概率值。

```
1: t ← 0
2: 初始化 Q(t) 3:
while t ≤ T do
    生成经典种群 C(t) 观察 Q(t) 4: 5: 6: 7: 8: 9: 10: 11:
    如果 t = 0 那么
        评估 C(t)
        P(t) ← C(t) 否
    则
        C(t) ← C(t) 和 P(t) 之间的重组
        评估 C(t)
        P(t) ← 选择(C(t), P(t)) 如果
12: 结束
15: end while 14: Q(t) ← 根据 P(t) 值更新 Q(t) 13: t ← t + 1
```

图 1:Q-NAS 算法。

在世代循环开始时,观察到 Q(t) 生成经典种群 C(t) (图 1 中的第 4 行)。每个量子个体可以产生不止一个经典个体。一旦准备就绪,就可以评估 C(t) (图 1 中的第 6 行)。评估过程涉及使用训练数据的子集对候选网络进行少量训练,并使用验证数据集为个体分配适合度分数。在第一代中,C(t) 个个体被排序并存储在 P(t) 中 (图 1 中的第 7 行)。

请注意,经典重组仅在第一代之后才有可能 (图 1,第 9 行)。

在后续世代中,我们已经有了一个排名 $P(t)$ 的种群,所以当我们生成新的经典个体时,我们必须选择哪些将留在 $P(t)$ 中,哪些将被新的取代 (图 11 中的第 11 行) 1). 在本文中,我们将分析两种选择机制: **稳态和精英主义**。第一种方法对新老种群进行排序,保留 k 个最好的个体;第二个替换除了最好的那个之外的所有旧个体。

完成循环后, $Q(t)$ 将根据 $P(t)$ 中的最佳个体进行更新 (图 1, 第 13 行)。这个想法是逐渐修改量子群体,以便它可以生成更接近最优的解决方案。换句话说,更新应该减少搜索空间并映射有希望的搜索区域。

当达到最大生成数 T 时,循环终止。进化完成后,最好的个体代表最终的架构描述。然后使用整个训练数据集对该网络进行再训练,并使用测试集将最终性能指标 (在本例中为准确性) 分配给该结构。该度量值用于将 Q-NAS 生成的模型的性能与其他作品 [14] 进行比较。

C. Q-NAS 人口

Q-NAS 的目标不仅是解决网络设计问题,还包括优化与训练过程相关的一些超参数 [14]。因此,Q-NAS 必须能够表示网络结构的分类空间和训练超参数的数值空间。

网络架构被编码为具有最大 L 个节点的链状结构。在这个结构中,每个节点都有一个与之关联的层 (或层块) 功能。

用户指定一个预定义函数列表,这将是每个节点的搜索空间;函数的名称被映射到整数。 [14] 中的作者在列表中包含一个“无操作” (NoOp) 函数,因此它们可以表示可变长度结构。经典染色体的网络部分只是一个整数数组,表示与每个节点相关的函数 [14]。

量子个体将为每个节点定义一个唯一的概率质量函数 (PMF)。考虑 N 个量子个体,最大网络大小 L 和大小为 M 的函数列表,量子种群将是一个形状为 (N, L, M) 的数组。所有节点的初始 PMF 都相同,但每个函数的概率值可以由用户定义。如果需要这种偏差,这使用户能够增加某些函数的初始概率。

获得经典个体的观察过程包括分别从每个节点的 PMF 中采样。

使用 Numpy [26] 离散采样。图 2 显示了此过程的示例,其中 $L = 3$ 且 $M = 5$ 。

在图 1 的第 13 行中,量子种群根据最佳经典个体进行更新。对于网络染色体部分,[14]中提出了一个简单的启发式方法。这个想法是增加节点中有希望的功能的概率并按比例减少其他概率。

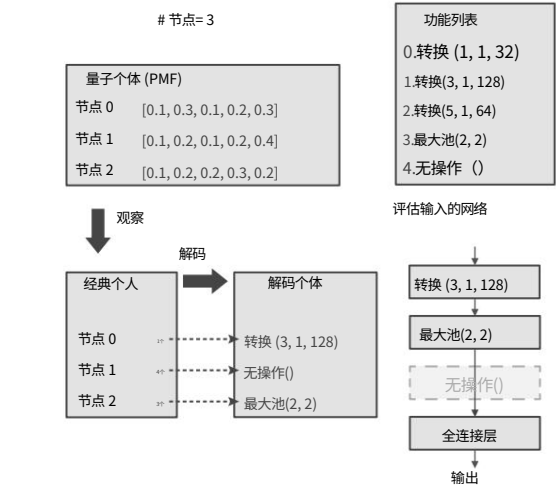


图 2: 网络生成过程。 $\text{Conv}(k, s, f)$ 是一个卷积层,内核大小为 $k \times k$,步长为 s ,过滤器为 f 。

观察过程从每个节点的 PMF 中采样。

解码过程将整数映射到函数名称。最终网络包括一个完全连接的 (FC) 分类器层。

染色体的第二部分使用基于 [17],[18] 的数值表示来编码一些训练超参数。对于每个要优化的变量 h_j ,根据其定义域定义概率密度函数 (PDF)。考虑一组 G 个数值超参数,量子染色体 q_i 是一个形式为 [14] 的数组:

$$[g_{i1} = p_{i1}(x), \dots, g_{iG} = p_{iG}(x)] \tag{2}$$

其中 $p_{ij}(x)$ 是 PDF,表示在观察时观察到特定值范围内的量子基因的概率。在 [14] 中,作者使用统一的 PDF,由其下限和上限 l_{ij} 和 u_{ij} 定义 h_j 可以假设的最小值和最大值。

再一次,观察过程包括从分布中抽样。在统一 PDF 的情况下,采样过程很简单,并使用其对应的 CDF $F_{ij}(x)$ 。它涉及以下步骤: (1) 在 $[0, 1]$ 区间内产生一个随机数 r ; (2) 找到使 $F_{ij}(x) = r$ 的 x ,这是采样值。由于 $F_{ij}(x)$ 是域内的一条直线, x 可由 [14] 得到:

$$x = r \times (u_{ij} - l_{ij}) + l_{ij} \tag{3}$$

仅在经典总体的数值部分,[14] 中的作者应用了传统的重组 (图 1 中的第 9 行),更具体地说是算术交叉运算符。

chromosome 的数字部分的量子更新是一种启发式算法,它也使用来自最佳经典算法的值。这种启发式将 PDF 转移到最佳值并根据当前经典值的范围减小它们的宽度。有关此更新的详细信息,请参见 [14]。

四、实验与讨论

[14] 中提出的第一个 Q-NAS 结果是我们新实验的起点。我们想通过两个主要目标来分析整个进化过程中的种群适应度。第一个是将 [14] 中使用的稳态选择机制与不太保守的精英主义选择进行比较。第二个验证 Q-NAS 是否可以在进化早期找到好的个体,以及它是否可以从添加早期停止机制中受益。

A. 实验一

第一个实验涉及将 [14] 中提出的运行的总体适应度与使用精英主义而不是稳态的新运行进行比较。我们使用与 [14] 相同的参数配置,为了清楚起见,我们将在此处重复它们。

Q-NAS 中的评估步骤包括使用可用数据的子集对候选网络进行训练和验证,如第 III 节所述。我们使用了与 [14] 中相同的 CIFAR 10 [21] 数据集方案,详情如下。

对于候选评估,原始 CIFAR-10 训练集的一个子集用于创建训练集和验证集。

我们随机选择每类 900 个样本进行训练,每类随机抽取 100 个样本进行验证,总共有 9000 张图像用于训练,1000 张用于验证。均值减法是唯一的预处理方法。对于数据扩充,我们使用随机翻转和零填充与随机裁剪相结合。

RMSProp 优化器用于训练过程 [14],它有两个超参数,称为衰减和动量 [27]。在 [14] 中,这些超参数使用了两种不同的配置,我们将在这里称为配置 1 和配置 2。第一个使用固定值的衰减和动量,如表 I 右侧所示。在配置中如图 2 所示,这些超参数是由 Q-NAS 使用染色体的数值部分连同权重衰减超参数一起进化而来的。总之,配置之间的差异仅依赖于超参数的演变。

在config 1中,没有进化超参数,只有网络结构,而在config 2中,三个超参数与网络结构一起进化。对于实验 1,我们仅使用配置 1,因为 [14] 中的最佳结果出现在这种情况下。

表 I (左)还显示了在 [14] 中选择并在此处保持相等的 Q-NAS 特定参数的值。节点数固定为 20,这意味着网络最多可以有 20 层。重复因子是每个量子个体将产生的经典个体的数量 在本例中为 4。更新世代参数指定量子更新过程将以更新量子速率参数定义的速率执行的频率。根据表 I 中的值,量子更新将每 5 代发生一次,速率为 0.1。

在第 III 节中,我们指出 Q-NAS 用户必须指定构成网络中每个节点的搜索空间的函数。表 II 显示了所选函数,我们可以将其分为三种类型:ConvBlock、Pooling 和 NoOp。ConvBlock 包含一个卷积层、一个批量归一化层和 ReLU 激活层。具体内核

表 I:实验的 Q-NAS 参数

进化参数		数值超参数		
		配置 1	配置 2	
世代 #	网络节点	衰变	0.9	[0.1, 0.999]
点 #	量子工业。重复更			
新世代更新量子率	5	势头	0.0	[0.0, 0.999]
	4 5			
		重量衰减	1.0e-4	[1.0e-5, 1.0e-3]
		学习率	1.0e-3	1.0e-3

大小、步幅和过滤器的数量列在表 II 中。对于 Pooling 层,我们有两个选项:最大池化和平均池化,它们的内核大小都是 2,步长都是 2。函数规范有利于简单性和计算成本效率 [14]。请注意,卷积层不会减少输入大小;尺寸缩减只留给池化层。如前所述,添加了 NoOp 函数,因此我们可以表示可有效长度的网络。

表二还提供了分配给每个功能的初始概率,按功能类型平均分配 (每个总计1/3)。

表二:层函数和初始概率

函数类型	内核大小	步幅	过滤器	操作。	初始概率
转换块	1*	1*	[32, 64]	-	0.042, 每个选项
	2*	1*	[32, 64, 128, 256]	-	
	3*	1*	[32, 64]	-	
池化	2*	2*	-	[最大值,平均值]	0.167, 每个选项
无操作	-	-	-	-	0.333

我们使用指定参数和配置 1 超参数重复五次进化运行。每个候选网络都在 K80 GPU 中进行评估,总计最多 20 个 GPU (与 [14] 中相同)。网络训练了 50 个 epoch,批量大小为 256;验证准确度是针对最后 5 个时期计算的。对于再训练阶段,我们使用所有可用的训练数据和两个 K80 GPU,持续 300 个时期。使用测试集获得最终精度。

表 III 显示了实验 1 的结果,其中我们详细说明了最终网络的大小及其在再训练阶段后在测试集上的准确性。在左侧,我们重复了 [14] 中配置 1 的值。表 III 的右侧显示了带有精英选择的新运行。请注意,在

就最终精度而言,选择方法似乎不会影响最终精度值。应该注意的是,来自稳态#4 的网络呈现出糟糕的结果,但其在演化阶段的准确度为 0.79 [14]。这意味着这个特定的 10 层网络没有足够的容量 [14]。然而,精英主义案例的演化时间更长。这可能表明稳态方法

正在引导朝着更快的训练结构方向发展,这些结构并不总是表现最好的。

表 III:实验 1 的结果

稳定状态			精英主义		
#	进化时间	再培训符合。	进化时间	再培训符合。	
	56 小时	13	0.8773	64 小时	14
1	52 小时	11	0.8812	60 小时	10
2	61 小时	12	0.8928	64 小时	14
3	54 小时	10	0.4925	62 小时	15
4 5	56 小时	15	0.8874	63 小时	14

图 3 显示了进化过程中最佳个体适应度和平均种群适应度,以实现稳定状态和精英主义的最佳运行。如果我们比较平均种群适应度,这两种方法之间的差异会变得更加明显。虽然均值增长相当缓慢,稳态情况下标准差很小,但最佳适应度值仍然有增加。在精英主义的情况下,最佳适应度曲线呈现出相似的特征,尽管群体适应度平均值的表现非常不同。

最佳适应度曲线的平台性质表明找到明显更好的结构具有挑战性。精英主义案例的平均适应度证实了这个想法。

在这种更加多样化的情况下(除最佳个体外,所有个体都被替换),会生成几个表现不佳的结构。换句话说,考虑到网络结构,人口并没有停滞不前;不同的架构呈现同样糟糕的结果,并且很难找到一个具有突出性能的架构。这个想法也可以解释稳态情况下平均适应度的缓慢增加:由于只选择最好的,新个体很难进入种群。总之,精英主义运行帮助我们理解了进化过程,但是

我们没有看到使用这种方法而不是稳态的明显优势。

B. 实验 2 在分析了

实验 1 中的适应度特征之后,我们定义了一个简单的提前停止标准:如果最佳个体适应度在 80 代内没有提高到 0.005 的阈值以上,则停止。第二个实验涉及在这个早期停止标准所指示的一代中重新训练最好的个体。我们应该强调,我们根据迄今为止在实验中观察到的行为选择了阈值和世代的值。我们在这里的目的只是验证一种简单的提前停止方法在 Q-NAS 中的适用性。

由于我们为进化运行保存了种群描述和适应度值,因此可以对保存的数据应用提前停止标准。这种方法允许我们比较最终网络和同一运行中的早期停止网络,从而减少随机种子和初始化问题的影响。我们将标准应用于 [14] 中的所有运行,并重新训练了指定的模型。

在表 IV 中,我们重复了左侧 [14] 中配置 1 和配置 2 的结果,包括最终精度、演化时间和网络大小。在表 IV 的右侧,我们展示了第 IV 节的提前停止标准所指示的最后一个模型的结果。右侧的 gen 列显示了如果 Q-NAS 被提前停止方法终止,它将会运行的世代数。我们在日志文件中使用时间戳来估计这些案例的演变时间。在表 IV 的某些行中,重新训练的准确性在添加早期停止后没有变化,这意味着它在已经找到最终模型时停止。这一观察结果证实了 Q-NAS 可能在进化过程的早期找到好的结构的观点。

在所有运行中,时间都显着减少,在最好的情况下(运行 2-3)显着减少了 75%。看待-

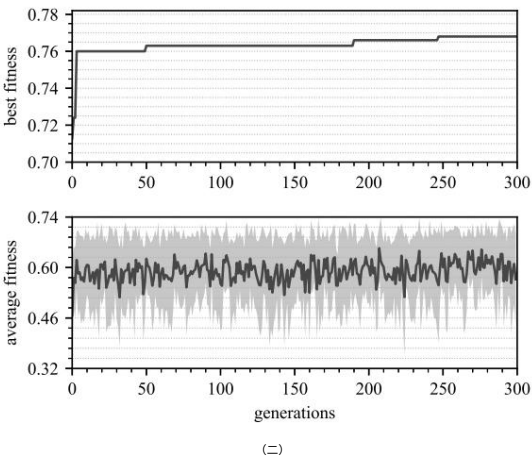
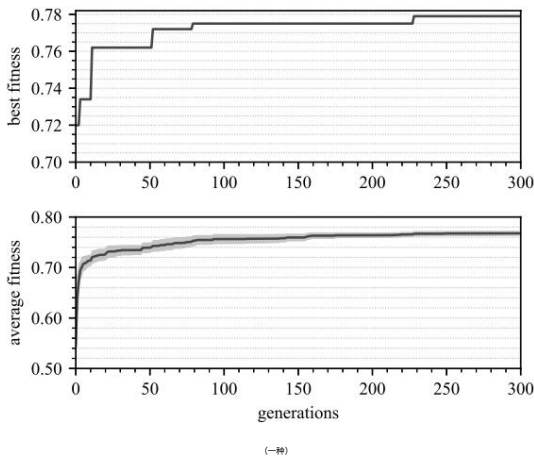


图 3:最佳个体适应度(顶部)和平均群体适应度,以及(a)稳态选择,运行 #3 和(b)精英选择,运行 #1 的标准差(底部)。

表 IV:实验 2 的结果

没有早停				带早停				
配置 -跑步	将军	进化。 时间	网 尺寸	再培训 符合。	将军	进化。 时间	网 尺寸	再培训 符合。
1-1	300	56 小时	13	0.8773	162	11	0.8812	88
1-2	300	52 小时	12	0.8928	123	10	0.8872	174
1-3	300	61 小时				25 小时	12	0.8912
1-4	300	54 小时				28 小时	10	0.4925
1-5	300	56 小时				31 小时	15	0.8874
2-1	300	52 小时	12	0.8765	90	11	0.8620	165
2-2	300	52 小时	16	0.8801	110	10	0.8620	86
2-3	300	54 小时				13 小时	16	0.8805
2-4	300	51 小时				14 小时	15	0.8575
2-5	300	51 小时				27 小时	17	0.8346

*创。 = 几代人 ,网络大小 = 最终网络层数

考虑到最终的网络性能,该方法是不够的,因为右边的所有模型 (有提前停止)与左边的模型 (没有提前停止)同样或几乎同样好,但是,需要进一步调查以验证该机制是否适用于其他数据集。此外,重要的是要检查标准中的阈值和世代是否是稳健的选择,否则,这可能会使该方法无效。

五. 结论

在这项工作中,我们扩展了之前对 Q-NAS 算法的分析,重点关注进化过程中适应度函数行为的评估。我们试验了增加种群多样性的精英选择机制。这种新的选择机制并没有以更快的方式提供性能更好的结构。

我们在最佳适应度曲线中观察到平台行为,这启发我们测试一种简单的提前停止方法,该方法仅考虑这些值的变化。早期停止机制的应用不仅减少了进化时间,而且保持了最终个体的质量。在某些情况下,我们能够在包含最终最佳模型的一代停止,花费原始时间的 1/4。我们的结果表明,添加这种简单的技术可以显著提高 Q-NAS 的效率。未来的工作涉及在其他数据集中测试相同的提前停止机制,以确认这一预期。

致谢

作者要感谢巴西机构 CNPq 和 FAPERJ 的财政支持。

参考

[1] A. Krizhevsky,J. Sutskever 和 GE Hinton, “使用深度卷积神经网络进行 ImageNet 分类”,神经信息处理系统进展第 25 期,2012 年,第 1097-1105 页。

[2] K. Simonyan 和 A. Zisserman, “用于大规模图像识别的超深卷积网络”,arXiv:1409.1556 [cs],2014 年 9 月。

[3] K. He,X. Zhang,S. Ren 和 J. Sun, “用于图像识别的深度残差学习”,arXiv:1512.03385 [cs],2015 年 12 月。

[4] Y. Lecun,L. Bottou,Y. Bengio 和 P. Haffner, “基于梯度的学习应用于文档识别”,JEEE 会刊,卷。 86, 没有。 11,第 2278-2324 页,1998 年 11 月。

[5] F. Hutter,L. Kotthoff 和 J. Vanschoren 编辑,自动机器学习:方法、系统、挑战。 Springer,2018 年,正在印刷中,可在 <http://automl.org/book> 上获取。

[6] B. Zoph 和 QV Le, “使用强化学习进行神经网络搜索”,arXiv:1611.01578 [cs],2016 年 11 月。

[7] B. Baker,O. Gupta,N. Naik 和 R. Raskar, “使用强化学习设计神经网络架构”,arXiv:1611.02167 [cs], 2016 年 11 月。

[8] Z. Zhong,J. Yan,W. Wu,J. Shao 和 C. Liu, “实用块智能神经网络架构生成”,2018 年 IEEE/CVF 计算机视觉和模式识别会议,2018 年 6 月,第 2423-2432 页。

[9] E. Real,S. Moore,A. Selle,S. Saxena,YL Suematsu,J. Tan,QV Le 和 A. Kurakin, “图像分类器的大规模演化”,第 34 届国际机器学习会议论文集,ser,机器学习研究论文集,卷。 70。

澳大利

亚悉尼:PMLR,2017 年 6 月,第 2902-2911 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”,arXiv:1802.01548 [cs],2018 年 2 月。

[11] B. Zoph,V. Vasudevan,J. Shlens 和 QV Le, “学习可扩展图像识别的可迁移架构”,2018 年 IEEE/CVF 计算机视觉和模式识别会议,2018 年 6 月,第 8697-8710 页。

[12] H. Cai,J. Yang,W. Zhang,S. Han 和 Y. Yu, “用于高效架构搜索的路径级网络转换”,第 35 届国际机器学习会议论文集,2018 年。

[13] H. Liu,K. Simonyan,O. Vinyals,C. Fernando 和 K. Kavukcuoglu, “高效架构搜索的层次表示”,国际学习表示会议,2018 年。

[14] D. Szwarcman,D. Civitarese 和 M. Vellasco, “Quantum-Inspired Neural Architecture Search”, 2019 年国际神经网络联合会议 (IJCNN),2019 年,接受出版。

[15] MD Platel,S. Schliebs 和 N. Kasabov, “Quantum-Inspired Evolutionary Algorithm:A Multimodel EDA”,IEEE Transactions on Evolutionary Computation,vol. 13,没有。 6,第 1218-1232 页,2009 年 12 月。

[16] K. Han 和 J. Kim, “用于一类组合优化的受量子启发的进化算法”,JEEE 进化计算汇刊,卷。 6,没有。 6, 第 580-593 页,2002 年 12 月。

[17] AVA da Cruz,M. Vellasco 和 M. Pacheco,用于数值优化的受量子启发的演化算法,柏林,海德堡:Springer Berlin Heidelberg,2007 年,第 19-37 页。

[18] , “应用于数值优化问题的量子启发进化算法”,JEEE 进化计算大会,2010 年 7 月,第 1-6 页。

[19] R. Stanley,Kenneth O 和 Miikkulainen, “通过增强拓扑进化神经网络”,进化计算,卷。 10,没有。 2, 第 99-127 页,2002 年。

[20] M. Vellasco,AVA Cruz 和 A. Pinho, “应用于神经建模的受量子启发的进化算法”,JEEE 世界计算智能大会,全体会议和特邀讲座,第 125-150 页,2010 年。

[21] A. Krizhevsky, “从微型图像中学习多层特征”,Tech。众议员,2009 年。

[22] R. Istrate,F. Scheidegger,G. Mariani,D. Nikolopoulos,C. Bekas 和 ACI Malossi, “TAPAS:架构搜索的无训练精度预测器”,arXiv:1806.00250 [cs],2018 年 6 月。

[23] M. Moore 和 A. Narayanan, “受量子启发的计算”,埃克塞特大学计算机科学系,埃克塞特,Tech。众议员, 1995 年。

[24] A. Narayanan 和 M. Moore, “受量子启发的遗传算法”,载于 IEEE 国际进化计算会议论文集,1996 年 5 月,第 61-66 页。

[25] MC Cardoso,M. Silva,MMBR Vellasco 和 E. Cataldo, “用于大气案例研究的尖峰神经网络的量子启发特征和参数优化”,Procedia Computer Science,卷。 53,第 74-81 页,2015 年。

[26] O. Travis E, “numpy 指南”,美国,2006 年。

[27] I. Goodfellow,Y. Bengio 和 A. Courville,深度学习,麻省理工学院出版社, 2016 年,<http://www.deeplearningbook.org>。