

神经网络的比较性能 通过反向传播和差分进化训练的变体架构

扎卡里亚欧萨勒姆	Rochan Avlur Venkat	贾纳维·马拉加瓦利	Arya Kumar Bhattacharya
计算机科学系	计算机科学系 计算机科学系 计算机科学系		
里尔中央理工学院	马恒达大学	马恒达大学	马恒达大学
法国阿斯克新城	印度海得拉巴	印度海得拉巴	印度海得拉巴
zakaria.oussalem@centrale.centralelille.fr	rochan170543@mechyd.ac.in	jahnavi160521@mechyd.ac.in	arya.bhattacharya@mechyd.ac.in

摘要 关于神经进化的工作最近倾向于关注不同类别神经网络的替代架构和超参数的设计,以使用进化算法(EA)提高性能和计算效率。主要地,学习权重参数的基本机制仍然是传统的反向传播(BP),现在被认为是深度学习(DL)的范例。DL的许多重要方面,如图像识别中跨层特征的层次结构、梯度消失等,都被认为是理所当然的,而没有意识到这些可能是由BP本身的特性隐式诱发的。与BP的局部梯度下降相反,执行全局优化的EA,如果广泛用于ANN训练,可能会破坏DL的这些假设方面,并构建替代的和有趣的观点。对于EA和BP之间的一对一性能比较,对于复杂的回归和分类问题,保持完全相同的架构、激活函数和数据集的研究活动令人惊讶地缺乏。这项工作部分填补了这一空白。结果表明,与标准BP解决方案相比,通过局部搜索增强的差分进化可以在嘈杂的工业数据集上产生略微更好的回归问题精度,并且在流行的图像分类数据集上具有相当的精度。因此,这为重新审视DL的上述假设方面建立了基线,并可能产生新的和有趣的观点。

反向传播(BP)作为基线机制,在某种程度上它可以被视为该领域的范例。

然而,构建ANN变体的架构和相关超参数的选择仍然是一项主要的手动练习,具有启发式但新颖的实验范围。基于梯度的优化在这里发挥的作用非常有限。遗传算法[2]和差分进化(DE)[3]等进化优化算法(EA)以及粒子群优化(PSO)[4]等类似的受自然启发的算法适用于同时搜索全局空间的候选解决方案群体,将探索与开发并列。重要的是,这些可以在数学上难以处理或不容易服从数学表示的优化场景中运行,例如ANN变体的架构、超参数的影响等。

因此,EA在优化ANN的这些方面得到了重要应用,这一领域被称为神经进化[5]。

进化算法除了在ANN架构和超参数的设计中发挥作用外,还可以应用于模型训练参数(即权重)的优化。然而,与主流的反向传播方法相比,这个角色被认为是次要的或微不足道的,并且对神经进化的引用几乎隐晦地假设仅使用EA设计架构和超参数。然而,EA对权重的优化可以对ANN、EA以及高性能计算(HPC)领域产生深远的影响,这是有非常重要的原因的。

索引词 神经网络架构、优化权重参数、反向传播、差分进化、局部交叉和变异、回归和分类、神经进化。

一、引言

人工神经网络(ANN)范围内的机器学习技术及其多种变体在过去十年中在研究和应用方面取得了显著扩展,通过基于梯度的算法训练模型参数所获得的效率在基础上实现了基础数据。事实上,早在1990年代引入的反向传播算法[1]仍然是ANN的标准训练主力。深度学习的所有主要进展、解释和解释都隐晦地假设

当应用于ANN以优化权重参数(即训练)时,这些方法遍历搜索空间而不对底层架构景观做出任何假设。这与本质上在本地搜索的梯度下降优化形成对比;当应用于ANN训练时,这转化为从最后(输出)层一直向后传播误差到输入层,沿途调制权重。

在主要由卷积神经网络 (CNN) 启用的图像处理应用程序的上下文中,现在公认的是早期层提取图像的低级特征,而后层开始识别更高级别的特征,最后识别对象[6-8]。尝试认识到这种对象识别的层次化进展在多大程度上是由训练的反向传播引起的是相关的。将其替换为基于 EA 的全局优化技术会改变层次识别模式吗?这能否导致更准确的识别,或者更高的效率反映在减少的层数或替代架构中?

此外,众所周知[9]、 [10]图像中对象的识别对特定像素的值非常敏感,并且这些像素的变化会导致识别的改变。

这种放大的灵敏度在多大程度上取决于基于梯度的训练。像 EA 这样的全局训练方法会导致更一致的识别吗?对可解释性[11]的相应影响是什么?此外,梯度消失和爆炸是与 BP 相关的众所周知的问题;基于 EA 的培训将如何影响与此问题相关的所有问题?例如,LSTMs [12]和 GRUs [13]被设计用来最小化和避免 BP 在循环神经网络训练中的这个问题。

要使用 EA 研究上述所有这些方面,首先使用 EA 开发经过训练的 ANN 变体模型是相关的,这些 EA 提供与 BP 训练模型相当精度的解决方案,用于完全相同的架构、激活函数和训练数据。这应该针对回归和分类类型的问题进行调查。这变得至关重要,因为如果 EA 被证明无法为相同的 ANN 生成与 BP 相当的精度的解决方案,那么上述调查的整个存在理由就变得有点站不住脚了。

此处报告的发展的主旨是使用 EA 和 BP 生成一系列架构以及分类和回归问题的解决方案,并研究它们的比较方式。经典和 CNN 架构都被考虑在内。因此,证明可以合成具有特定算法特征的 EA,从而生成与 BP 一样准确的解决方案。

有人可能会质疑,考虑到上述探索性命题的高影响价值,是什么阻止了分析师和研究人员更早地深入研究它们?答案是双重的。首先,任何基于 EA 的解决方案在训练阶段都需要大约 $N \times \text{BP-baseline}$ 的计算量,其中“N”是种群中候选者的数量, BP-baseline表示计算时间收敛的 BP 解决方案。即使使用 BP,高级 ANN 变体的深层架构也需要大量计算时间。合理数量的候选人 (如 60-200)会将其放大两个数量级,这在计算上是巨大的。目前的作者开发了[14]新型混合 CPU-GPU 计算范例,利用 EA 算法的自然并行性,将基于 EA 的解决方案的计算时间减少了大约 400 倍,从而使拟议的研究进入可行性领域。人们可能只是注意到,当训练有素的 ANN 输入

进入生产阶段,权重最初是如何学习的并不重要,计算时间将完全相同。

其次,神经进化的趋势一直是设计架构和超参数,以提高解决方案的准确性和时间;与 BP 就相同的架构和超参数进行一对一的比较几乎不是人们感兴趣的课题。也许一个隐含的假设盛行,即 BP 总是会变得更好?本研究打消了这种观点,为沿着上述课题进行分析扫清了道路。

在这项工作中实施的 EA 是差分进化[15]的特定版本,由作者开发并发现比其他 EA [16]工作得更好。Vesterstrom 和 Thomsen 进行的一项早期研究表明,DE 往往优于 PSO [17]。上述全局 DE 通过 Zhang 和 Xu [18]的局部搜索功能得到增强,并在本研究中用于与 BP 对变体 ANN 架构和问题进行一对一比较。

本文的其余部分安排如下。第二节提及现有工作,有助于将当前的发展置于背景中;第三节概述了当前的 DE 公式并解释了与本地搜索的融合。第四节介绍了不同的体系结构以及 BP 和 EA 解决方案之间的相应比较。第五节讨论结论和进一步的工作。

二.相关工作

使用 EA 优化神经网络的想法在过去已经被探索过,通常被描述为神经进化[5]。近年来,EA 作为计算上可行的方法一直在稳步发展:(1) ANN 架构的自动优化[19-21]和 (2) ANN 的训练,后者在本文中进行了探讨

纸。

在 1990 年代和 2000 年代初期使用 EA 训练 ANN 的令人鼓舞的结果激发了一些研究人员最近探索这个令人兴奋的交叉点[22-24]。为了简单和与这项工作相关,我们将有关使用 EA 训练 ANN 及其变体的文献大致分为两类。一方面,我们有使用差异进化 (DE) 的方法。在第二组中,我们有所有其他变体,例如粒子群优化 (PSO) 等。已经探索了来自这两个组的算法来训练 ANN。例如,在[25]中, PSO 用于成功训练 CNN,在 MNIST 数据集上达到 95% 的准确率。其他人提出了自己的编码策略和修改 PSO 进行多目标优化以执行图像分类任务[26]。在[22]中,提出了一种无梯度方法,通过使用简单的遗传算法 (GA) 来进化 CNN 的权重,并使用一组固定长度的染色体进行强化学习。

同样,使用 DE 训练 ANN 的尝试在流行的基准测试数据集上得到了很好的记录和评估。在[27]中,经典 DE 用于训练在 MNIST 数据集上达到 88% 准确率的 ANN。在此基础上, [27]的作者提出了一种新的 DE 自适应版本,称为 MAB ShaDE ,改进了他们之前的实现,在 MNIST [28] 上实现了 89.4 - 90.4% 的准确率。以上尝试,

然而,将图像展开成一个长数组并使用简单的线性前馈 ANN 架构。在 [29]中已经探索了使用 DE 训练 CNN,通过使用混合方法来完成图像分类任务。在这项工作中,如第 1 节所述。1,我们使用了 DE 的修改版本,该版本增强了本地搜索功能。

三、算法和变体

如上所述,我们正在使用**差分进化 (DE) 的变体进行全局搜索并将其与局部搜索相结合**。第一个小节介绍了此处使用的全局 DE。第二个介绍了本地搜索方法。

A. 基线全局差异进化

形式上,若解空间的维数记为D,候选解的个数 (即种群大小)为N,则第G代解 X_i, G 的第i个候选解的元素可记为

$$X_{i,G} = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\} \quad (1)$$

对于所有 $i = 1, \dots, N$ 。

DE 过程从根本上从当前候选集中生成新的解决方案,方法是将两个随机选择的候选解决方案向量之间的加权差异添加到第三个以生成“突变”向量,然后在现有向量和“突变”向量之间创建交叉被称为“试验”载体。后者只有在被发现更“适合”时才被允许替换现有向量。这种“适合度确定”练习的复杂性完全取决于所考虑问题的性质。

若 V_i, G 表示突变向量,则根据称为 DE/rand/1 [3] 的基线 DE 过程

$$V_{i,G} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G}) \quad (2)$$

其中 r_1, r_2 和 r_3 是小于 N 的随机整数,彼此不同且与 i 不同, F 通常介于 0.5 和 1 之间。这个基线过程有很多变体,有时会考虑两个而不是一个差项,人口中的最佳解决方案已纳入等;替代方案的描述可以在 [3, 30] 等中看到。

在“突变”向量 V_i, G 和目标向量 X_i, G 之间执行交叉,以根据以下公式生成“试验”向量 Z_i, G

$$Z_{i,j,G} = \begin{cases} V_{i,j,G}, & \text{如果随机数 } (0,1) \leq Cr \\ X_{i,j,G}, & \text{否则} \end{cases} \quad (3)$$

其中 $Z_{i,j,G}$ 是试验向量 Z_i, G 的元素 j, $randj(0, 1)$ 表示应用于元素 j 的 0 和 1 之间的随机数, Cr 是交叉阈值,通常设置在 0.4 和 1 之间。

在最后的步骤中,通过直接一对一比较,在更高适应度的基础上,在 Z_i, G 和 X_i, G 之间选择下一代候选者 i。

在目前的工作中,突变向量是根据替代方案生成的 (在 [3] 中提出并且当前也使用

[31, 32]中的作者发现它比其他的更好 DE 变体)

$$V_{i,G} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G}) \quad (4)$$

其中 R 设置为 0.5, F 跨代在 -2 和 +2 之间随机变化 (对于一代内的所有都相同)。等式中的交叉概率 Cr 。(3) 设为 0.9。

B. 本地搜索的修改

对 Zhang 和 Xi [18]提出的局部搜索变体进行了修改,特别是与变异相关的方面,然后与我们的全局搜索 eqs 相结合。(3-4),生成在这项工作中广泛使用的工作优化算法。

任何一代 G 的局部搜索过程都按顺序进行局部交叉和局部变异。交叉步骤根据它们各自的目标值 $f(X_i)$ 对所有候选 X_i 进行降序排序 (对于最小化问题)。因此,对于两个解决方案 X_i 和 X_j , 其中 $j = 1, \dots, N$, X_i 放在 X_j 之前

$$f(X_i) < f(X_j) \quad (5)$$

接下来,遍历这个有序的候选序列,并在当前候选和它的下一个序列之间生成一个中间临时候选,方法是在每个维度中简单地在两者之间进行插值。这表示为

$$X_{i+1,G} = X_i + r(i)(X_j - X_i) \quad (6)$$

其中 $r(i)$ 是介于 0 和 1 之间的随机数。

如果中间候选人的适应度高于原来一对中更好的一个,然后它取代 X_i 。因此

$$f(X_{i+1,G}) < f(X_i) \quad (7)$$

在完成对所有对的遍历后,候选者再次按降序排列以进行变异步骤。

对于变异,首先确定所有候选 X_i 中每个维度的下限和上限,并表示为 L_j 和 U_j 。为突变选择的候选者在一定程度上从其在 D 维空间中的位置向该空间中的 U 或 L 点移动。基于两个考虑因素,突变的概率在候选人和世代之间有所不同。首先,“适应性较差”的候选者,即排序后 f 值较小的那些,比适应性较好的候选者发生更多的变异,因为与适应性好的候选者相比,人们更愿意“探索”适应性差的候选者。这反映在图 5 中算法中所示的 pm 值中。1.

其次,突变的概率在特定用户定义的超参数 G_C 中跨代循环变化,如图 2 所示。1.用于计算变量 q (定义在图中)。这种概率在周期开始时很高,在接近尾声时下降到接近零。将转移后的候选人的适应度值与其转移前的值进行比较

如果发现候选人的适应性更好,则候选人将被安置在新职位上。这在算法上表示如下:

最小化问题的局部变异算法,在任何一代 G

```

大小 ← j 分钟 [ , 1, 2, ..., N ], 1, ..., j
j ← 最大 [ , 1, 2, ..., N ], ∀ ∈ j
GC ← i = 1, ..., N do (exp 2) 突变参数的循环变化范围

下午 N 我 N 1/ # 突变概率
q = - × ( GC sig( ) sig( ) 表示 -
sigmoid 函数 for j = 1, ..., D do r, b ← rand(0, 1),
randint(0, 1) if b == 0 then

XXU 请求下午 ( ), j. 我
# . 表示标量乘法
别的
XXL 请求下午 ( ), j.
结束于
如果 (f) ≤ 温度 X 那么
←
结束于
```

图 1. 局部搜索下的变异算法,在 G 代。
在这项工作中考虑的变异参数的周期变化范围是50。

四、BP和DE解决方案的精度比较

本节介绍了使用反向传播 (BP) 获得的解决方案与通过局部搜索 (DE) 增强的差分进化获得的解决方案之间的比较,这些解决方案是根据第 2 节中的讨论执行的。1. 这意味着在相同的架构、激活函数和训练数据之间进行比较,但学习机制要么是 BP 要么是 DE。

首先,我们展示两个回归数据集的结果,然后展示 MNIST 分类数据集的结果。

回归案例 1 – 来自嘈杂工业的预测数据

第一个回归数据集与钢铁制造设施中通过光整机 (SPM) [33]轧制的钢带的屈服强度预测有关。SPM 是钢铁制造工艺链的最后阶段,在轧制过程中对薄钢带施加小的伸长率,从而改变其屈服强度。每个数据样本 (即一个条带) 由 13 个输入变量和 1 个输出变量组成。输入包括整个链条中累积的各种参数,重点是最终冷轧阶段的参数,输出当然是屈服强度。输入之一是 SPM 中赋予的伸长百分比。在五年的运行期间,总共积累了 8000 个这样的样本。所有数据,除了作为控制设定点的伸长百分比 (这是第 8 个输入变量)之外,都是

从传感器累积,不用说,存在显着的测量误差,表现为噪声。

训练 ANN 以在将数据按 80:20 进行训练:验证后预测屈服强度。ANN 架构如图 1 所示。2. 参数总数为 3301,包括偏差。该 ANN 由 BP 和 DE 独立训练。在这两种情况下,相同的数据样本被分配到训练和验证子集中。

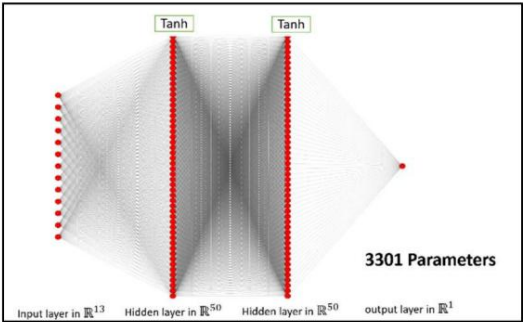


图 2. 回归问题 1 和 2 的 ANN 架构。

对于 BP 训练过程,我们使用 32 的小批量大小,使用Pytorch库中默认训练参数的 Adam 优化,即学习率参数 0.001,beta1 和 beta2 分别为 0.9 和 0.999,以及 epsilon 1.0e-7。

对于 DE 训练过程,我们使用的人口规模为 200,交叉概率为 0.9,等式中的 R 值。(4) 作为 0.5,同一方程中的 F 值在每一代的所有候选人中在 [-2, 2] 范围内随机变化。此外,我们将权重上限和下限引入为 [-30, 30],即训练参数的异常值被限制在各自的范围内。循环值GC如图5所示。1 设置为 50。

图 3 比较了 R 平方值,因为它随着时代 (对于 BP)或世代 (对于 DE)而演变。该值是通过将预测值与验证样本的实际值进行比较而获得的,因此如果所有预测样本值都恰好位于从实际值中提取的回归线 (此处为线性)上,则将获得 R 平方 1,并且随着预测值从回归线的分散增加,R 平方将按比例减少。可以看出,DE 解得到的最好的 R 平方约为 0.84,比 BP 解得到的最好的 R 平方要好 0.007,即DE 为这个回归问题提供了稍高的精度。

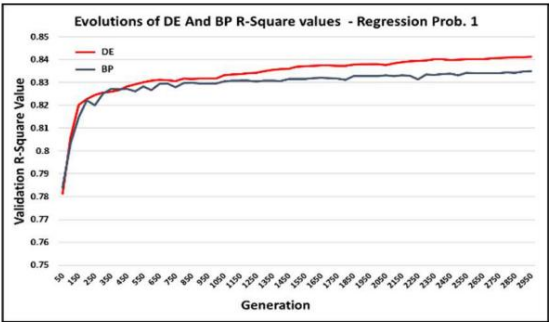


图 3.针对回归问题 1 的 DE 和 BP 训练,跨代以R2值表示的准确性增加。

无花果。图 4 和图 5 比较了验证数据集和训练数据集之间分别来自 BP 和 DE 的 R 平方值的演变。可以看出,验证数据始终具有比训练更高的准确性,这是违反直觉的,并且是数据中噪声破坏整体准确性的特征。图 6 将预测的屈服强度与实际测量值进行了比较,来自 BP (与 DE 非常相似)。这说明并解释了所讨论的 R 平方值;实际值在 x 轴上,预测值在 y 轴上。

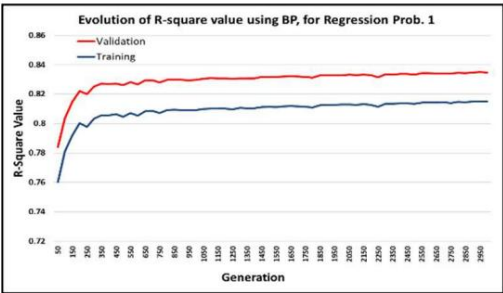


图 4. BP 训练的回归问题 1、训练和验证的R2演变数据集。

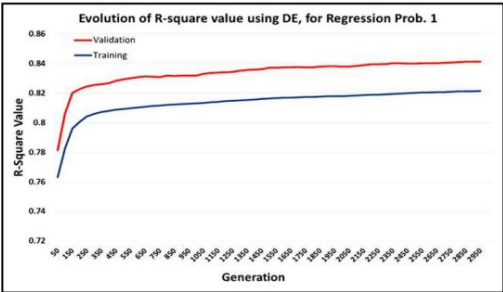


图 5. DE 训练的回归问题 1、训练和验证数据集的R2演变。

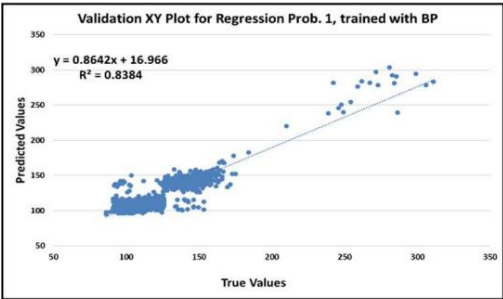


图 6. 用 BP 训练的回归问题 1 输出的最终预测值与真实值,反映了最终的R2 值。DE 的图非常相似 (未显示), R2值为 0.8426。

B. 回归案例 2 – 嘈杂工业过程的控制

这里的数据与案例 1 中使用的数据完全相同,不同之处在于屈服强度预测问题转化为 SPM 伸长率设定点控制问题。

这是通过简单地将第 8 个输入变量 (数据集中的第 8 列)与输出变量互换来创建的,因此我们不是预测给定伸长率设定点的屈服强度 (所有其他变量对每个样本保持完全相同),而是如果要达到一定的屈服强度,则尝试计算伸长率设定点应该是多少,即控制伸长率 (作为 ANN 输出)以获得条带的特定屈服强度, (作为 ANN 输入输入)当所有

其他参数 (即输入)是已知的。

这里需要注意的重要一点是,我们使用了与案例 1 中完全相同的架构,并且为 BP 和 DE 优化使用了完全相同的超参数。图 7 比较了来自 BP 和 DE 验证数据集的R2的演变;可以看出, DE 的准确度 (R2) 为 0.97,而 BP 的准确度 (R2) 大约低 0.004,这意味着 DE 在这种情况下也产生了更高的准确度。

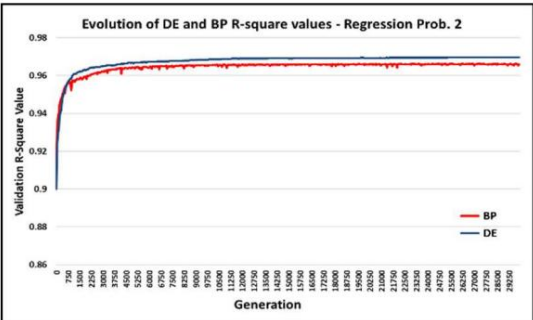


图 7.针对回归问题 2 的 DE 和 BP 训练,跨代以R2值表示的准确性增加。

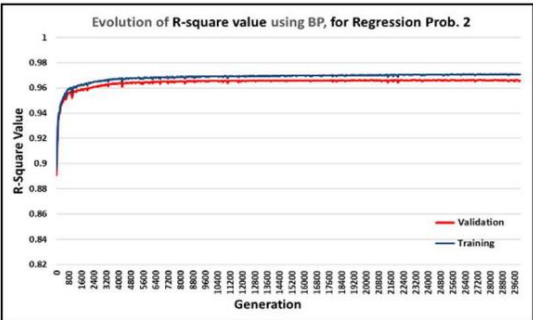


图 8. BP 训练的回归问题 2、训练和验证数据集的R2演变。

有人可能会质疑为什么在这个案例中获得比之前高得多的 R 平方,这将在下面简短地解决。无花果。图 8 和图 9 比较了 BP 和 DE 训练的验证数据集的准确度演变,这里可以看出遵循训练应该更准确的直觉规则。图 10 通过绘制计算值与实际值来说明实际的 R 平方。一个重要的观察是输出实际值位于 x 轴上的特定点,这是因为这个 - 伸长百分比 - 是根据特定点 (域)的某些规则设置的值,并且不会在整个点上任意变化整个范围,因此与之前的输出是屈服强度并用噪声误差测量的情况不同,这里的误差或噪声为零。由于 ANN 输出无噪声,因此总体精度比案例 1 高得多。

有趣的是,虽然 DE 训练的案例在这些单输出回归案例中表现出略微优越的性能,但当输出数量增加到 2 及以上时,DE 性能会下降。这是因为全局 DE 过程试图由 2 个或多个误差分量组成的单个成本函数一起调制所有权重,而局部 BP 过程调制权重以满足每个输出节点的误差

本地。对这一观察有两种可能的反应,一种是,当处理具有m个输出的回归问题时,可以并行创建m个

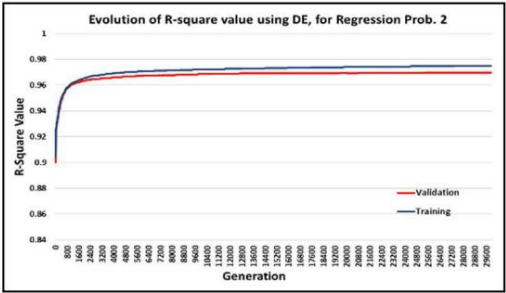


图 9. DE 训练的回归问题 2 的R2演化,训练和验证数据集。

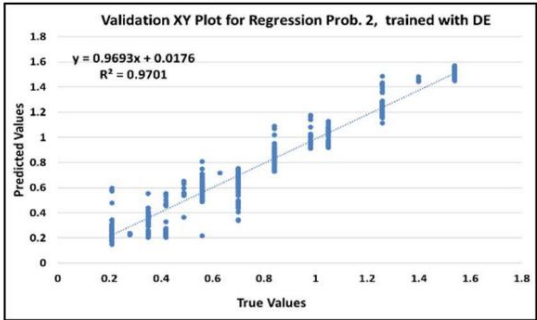


图 10. 使用 DE 训练的回归问题 2 输出的最终预测值与真实值,反映了最终的R2值。非常相似的 BP 图 (未显示), R2值为 0.9662。

每个 ANN 都有一个输出节点,同时独立训练。其次,可以研究具有与输出节点数量一样多的目标的多目标 DE (m需要非常小),然后在收敛的 Pareto 曲面上与曲面所有角点等距的点进行权衡。

可能会注意到,在回归案例中,我们仅使用了经典的全连接 ANN;在接下来的分类问题中,我们同时使用了经典的 ANN 和 CNN。

C. 比较 MNIST 上的分类性能数据集

使用标准 MNIST 数据集首先训练 ANN,然后使用 BP 和 DE 方法训练 CNN,并比较两种优化方法在验证子集上获得的最终精度。

ANN 上 BP 训练的超参数与两个回归案例相同,但有以下两个区别 小批量大小为 64 并用 Adagrad 代替 Adam 进行优化。对于 Adagrad,初始累积值为 0.1,学习率为 0.001,epsilon 为 1.0e-07。对于 CNN,minibatch 大小更改为 16,其他所有内容保持不变。

用于 ANN DE 训练的超参数与回归案例中的完全相同。然而,对于 CNN,在放弃突变因子pm的适应性依赖概率 (见图 1)后可以获得更好的结果,即 pm

总是取为 1。

图 11 说明了用于 ANN 的体系结构和激活函数 (回想一下这两种优化方法是相同的)。参数总数为 11250。图 12 显示了 BP 训练下训练和验证数据集的准确度跨时期的变化。可以看出,验证集的最高准确率 95.4% 是在不到 1000 个 epoch 的情况下实现的。图 13 显示了在 DE 训练下准确率的相应演变。可以观察到,验证准确度仅在大约 15,000 代时达到最高水平,尽管它们几乎在大约 10,000 代时趋于稳定。最高值为94.8%。因此,对于这种具有完全连接的 ANN 架构的分类案例,BP 训练不仅产生比 DE 略高的准确度,重要的是,BP 学习的解决方案收敛得更快。接下来我们转向 CNN 架构。

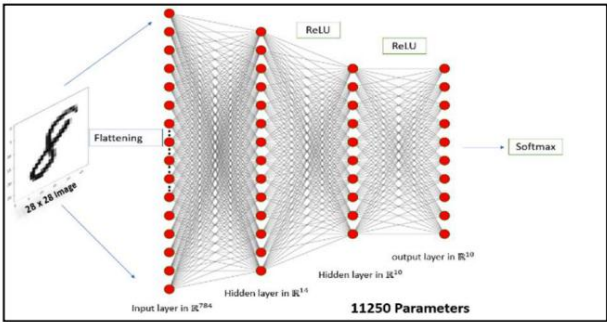


图 11. MNIST 分类问题的 ANN 架构。

图 14 显示了用于 CNN 的架构。这是一个相当简单的设计,避免了任何池化和零填充,最重要的是,完全避免了全连接的预最终层,否则众所周知,它可以提高精度水平但会增加参数的数量。这有助于将参数数量保持在仅 6740 个的低水平。图 15 显示了通过 BP 训练获得的准确度,在 500 个时期内获得的最高验证准确度为 98.1%,尽管该值稳定在少于该时期数的一半。

图 16 显示了通过 DE 训练获得的相应精度。在 20,000 代之后达到了 97.4% 的最高验证精度值,尽管这个值实际上是在大约 10,000 代时获得的,此后趋于稳定。

这再次表明,对于分类,BP 给出了稍高的准确度,而且达到的速度要快得多。

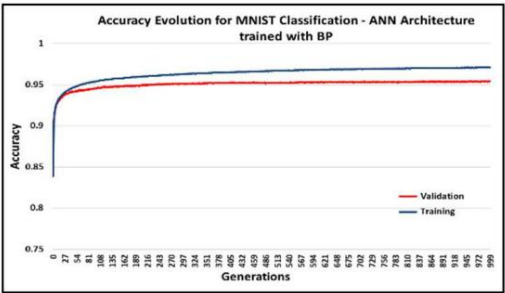


图 12. BP 训练的 MNIST 分类问题在 ANN 架构、训练和验证数据集上的精度演变。

结果是一些观察结果。首先,那

CNNs 即使参数数量几乎减少一半,在图像分类问题上产生的准确度也比 ANNs 高出约 2.5%,与训练方法 (即 BP 或 DE)无关。这在当地是可以理解的

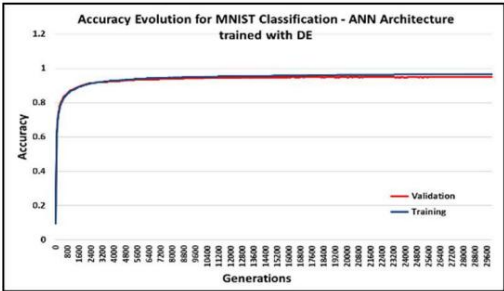


图 13. DE 训练的 MNIST 分类问题在 ANN 架构、训练和验证数据集上的精度演变。

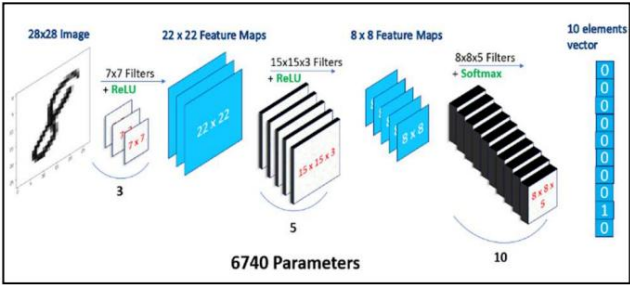


图 14. MNIST 分类问题的 CNN 架构。

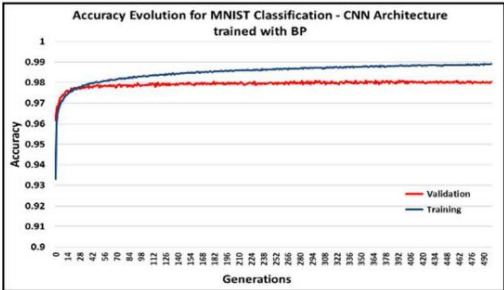


图 15. BP 训练的 MNIST 分类问题在 CNN 架构、训练和验证数据集上的精度演变。

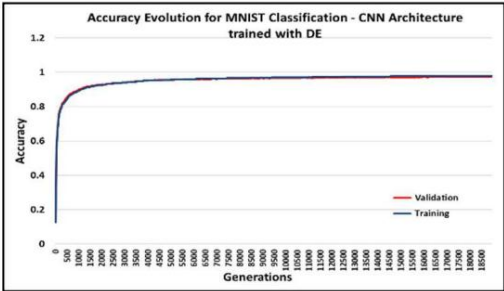


图 16. DE 训练的 MNIST 分类问题在 CNN 架构、训练和验证数据集上的精度演变。

图像的邻里关系特征被恰当地映射到 CNN 设计的哲学中。其次,无论架构如何,我们使用 DE 训练在 CNN 上获得的 97.4% 准确度是作者迄今为止在 EA 训练的神经网络中看到的最好的。

对替代 CNN 架构进行了调查

它们在最后两层中过滤器的大小以及参数总数上有本质的不同。“轻型”架构如图 1 所示。14,表 1 列出了重要的架构参数。表 2 和表 3 提供了有关替代架构的信息,“标准”架构具有 9180 个参数,“重型”架构具有 21210 个参数。所有架构的输入和输出大小都相同。

表 1. 具有 6740 个参数的 “Lighter”CNN 的架构数据,这也在图 1 中进行了说明。14.

层数	数过滤器 (通道)= Num.特征图	过滤器尺寸 (始终为正方形)	数外部零填充
1 st	3 st	7	0
2 nd	5 th	15	0
3 rd	10	5 th	0

表 2. 具有 9180 个参数的 “标准”CNN 的架构数据

层数	数过滤器 (通道)= Num.特征图	过滤器尺寸 (始终为正方形)	数外部零填充
1 st	3 st	7	0
2 nd	5 th	11	0
3 rd	10	12	0

表 3. 具有 21210 个参数的 “较重”CNN 的架构数据

层数	数过滤器 (通道)= Num.特征图	过滤器尺寸 (始终为正方形)	数外部零填充
1 st	3 st	7	0
2 nd	5 th	11	5 th
3 rd	10	18	0

图 17 显示了 BP 案例的准确度变化,从这里可以看出,“重”架构在大约 20 个时期内达到了 99% 的最高验证准确度,并且随着参数数量的减少,准确度也下降了随着时代数量的增加而获得。

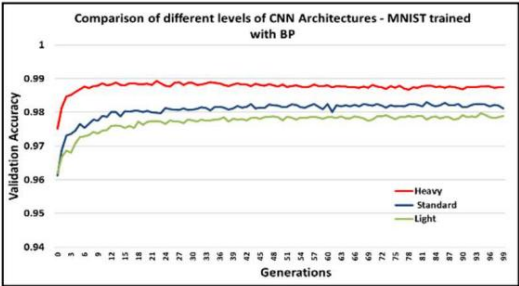


图 17. 应用于 MNIST 分类、使用 BP 训练的不同 CNN 架构的跨时期准确度变化。

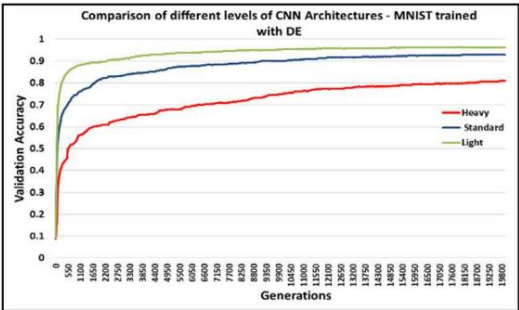


图 18. 应用于 MNIST 分类、使用 DE 训练的不同 CNN 架构的跨时期准确度变化。

值得注意的是,对于图 4 所示的 DE 训练解决方案,观察到的情况恰恰相反。18. 在这里,较轻的架构获得了最高的精度,随着参数数量的增加,精度实际上下降了,但即使在 20,000 代之后也没有达到收敛。因此,有人可能会争辩说,如果可以研究更轻的架构,则可以在更少的世代数下获得更高的精度。

五、结论和进一步工作

这项工作的目的是比较进化算法,以通过局部搜索增强的差分进化为代表,与反向传播作为不同神经架构权重训练的竞争机制并分析它们的性能。应用这两种机制的架构、激活函数和数据集需要相同才能进行比较,并且需要考虑回归和分类问题。

更远的目标是将 EA 训练提升到一个阶段,其中许多方面通常假设隐性特征是深度学习的特征,但这里推测是 BP 训练的横向表现,可以用不同的视角进行研究。这些目标已经成功实现,并且表明 DE 在回归问题和分类问题上的表现都非常出色。

明显的扩展是追求更深层次的目标并调查第 1 节中讨论的不同方面。1. 另一个工作方向是结合Neuro-Evolution,使用Multi-Objective Optimization同时改进架构和超参数,以及权重,以最小化计算时间和最大化准确性为双重目标。图 1 中讨论的观察结果。18 激励朝这个方向努力。

参考

[1] DE Rumelhart,GE Hinton and RJ Williams, “通过反向传播错误学习表征”,《自然》,第 1 卷。323,没有。6088,1986,第 533–536 页。

[2] JH Holland 等人,自然和人工系统中的适应:对生物学、控制和人工智能的应用的介绍性分析,麻省理工学院出版社,1992 年。

[3] R. Storn and K. Price, “差分进化 一种简单有效的连续空间全局优化启发式”,《全局优化杂志》,卷。11,没有。4,1997,第 341–359 页。

[4] J. Kennedy 和 R. Eberhart, “粒子群优化”,载于 ICNN 95 国际神经网络会议论文集,卷。4。IEEE,1995 年,页数 1942–1948。

[5] Edgar Galván 和 Peter Mooney, “深度神经网络中的神经进化:当前趋势和未来挑战”,arXiv:2006.05415,2020 年。

[6] MD Zeiler 和 R. Fergus, “可视化和理解卷积网络”,欧洲计算机视觉会议。斯普林格,2014 年,第 818–833 页。

[7] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs 和 H. Lipson, “通过深度可视化了解神经网络”,arXiv 预印本 arXiv:1506.06579,2015 年。

[8] A. Nguyen, J. Yosinski 和 J. Clune, “多方面特征可视化:揭示深度神经网络中每个神经元学习不同类型特征”,arXiv 预印本 arXiv:1602.03616,2016 年。

[9] K. Simonyan, A. Vedaldi 和 A. Zisserman, “深度卷积网络内部:可视化图像分类模型和显着图”,arXiv 预印本 arXiv:1312.6034,2013 年。

[10] W. Samek, A. Binder, G. Montavon, S. Lapuschkin 和 K.-R. Müller, “评估深度神经网络所学内容的可视化”,IEEE Trans. 关于神经网络和学习系统,第一卷。28,没有。2016 年 11 月,第 2660–2673 页。

[11] G. Vilone 和 L. Longo, “可解释的人工智能:一个系统的审查”,arXiv 预印本 arXiv:2006.00093,2020 年。

[12] S. Hochreiter 和 J. Schmidhuber, “长短期记忆”,神经网络,卷。9,没有。8,1997,pp. 1735–1780。

[13] K. Cho 等人, “使用 RNN 编码器解码器学习短语表示以进行统计机器翻译”,arXiv:1406.1078,2014。

[14] Rochan Avlur, Zakaria Oussaleh 和 Arya K. Bhattacharya, “在混合 CPU-GPU 架构上使用并发任务分配训练具有差分进化的卷积神经网络”,IEEE 进化计算大会 (CEC),2021 年。

[15] R. Storn 和 K. Price, “DE - 一种简单有效的连续空间全局优化自适应方案”,技术报告,卷。25,没有。6,1995,第 95–102 页。

[16] AK Bhattacharya 和 D. Sambasivam, “钢铁制造连铸过程中振荡参数的优化:遗传算法与微分进化”。奥地利维也纳 InTech,2009 年,第一卷。572。

[17] J. Vesterstrom 和 R. Thomsen, “差分进化、粒子群优化和进化算法在数值基准问题上的比较研究”,2004 年进化计算大会论文集,卷。2。IEEE,2004 年,页数 1980–1987。

[18] J. Zhang 和 J. Xu, “不连续优化问题的新微分进化”,第三届国际自然计算会议,ICNC 2007,卷。3。IEEE,2007 年,第 483–487 页。

[19] Thomas Elsken, Jan Hendrik Metzen 和 Frank Hutter, “通过拉马克进化进行高效多目标神经架构搜索”,arXiv:1804.09081,2018 年。

[20] 刘汉孝等。 “高效架构搜索的层次表示”,国际学习表示会议,2018 年。

[21] Risto Miikkulainen 等人,进化的神经网络,Ch. 15 in 人工智能在神经网络和脑计算时代,学术出版社,2019。293–312。

[22] Felipe Petroski Such 等人, “深度神经进化:遗传算法是训练深度神经网络进行强化学习的竞争性替代方案”,arXiv:1712.06567,2017 年。

[23] K. Pawe czyk, K. Michal 和 N. Jakub, “基因训练的神经网络”,遗传和进化计算会议论文集,2018 年。

[24] Chrisantha Fernando 等人, “进化卷积:可区分模式生成网络”,遗传与进化计算会议论文集,2016 年。

[25] Arie Rachmad Syulistyo 等人, “用于卷积神经网络 (CNN) 训练优化的粒子群优化 (PSO)”,Jurnal Ilmu Komputer dan Informasi vol. 9,没有。2016 年 1 月,第 52–58 页。

[26] Bin Wang 等人, “通过用于图像分类的多目标粒子群优化进化深度神经网络”,遗传与进化计算会议论文集,2019 年。

[27] M. Baiocchi 等人,学习大型神经网络的微分进化。技术代表,可在 https://github.com/Gabriele91/DENN_RESULTS-2018,2018 获取。

[28] M. Baiocchi, GD Bari, A. Milani 和 V. Poggioni, “神经网络优化的微分进化”,数学,卷。8,没有。1,2020, p. 2020 年 69 号。

[29] Bin Wang 等人, “一种设计用于图像分类的深度卷积神经网络的混合差分进化方法”,澳大利亚人工智能联合会议,Springer, Cham, 2018 年。

[30] S. Das 和 PN Suganthan, “差异进化:最先进的调查”,IEEE 进化计算交易,卷。15,第 1 期,2011 年 2 月,第 4–31 页。

[31] AK Bhattacharya, D. Aditya 和 D. Sambasivam, “使用差分进化估算操作高炉反应堆不可见内表面”,应用软计算,卷。13,2013 年 5 月,pp. 2767–2789。

[32] SR Gautam, M. Jahnavi, P. Thangeda 和 AK Bhattacharya, “使用多目标进化算法合成战术空中交战中的最佳轨迹”,多学科分析和优化的进展,机械工程讲义, Springer Singapore, 2021 (在新闻)。[33] http://www.sms-siemag.com/download/W6_7_304E_Skin_Pass_Mills_References.pdf, 最后访问时间为 2021 年 4 月 18 日。