基于 MetaFormer 的高效神经架构搜索

张嘉淏 [1] **刘添羽** [1]

1 华东师范大学计算机科学与技术学院 上海市 200062

（10185102243@stu.ecnu.edu.cn）

摘 要 Transformer 是一种基于自注意力机制的神经网络架构。近年来，随着大量新型自注意力机制的涌现，这一架构在计算机视觉任务中得到广泛的应用。然而，近期研究成果表明，Transformer 在视觉任务上的有效性由整体结构而非自注意力机制决定，在 Transformer 中将自注意力机制替换成其他模块并不会损害其特征提取能力。基于这一事实，本文提出了一种基于 MetaFormer 的神经架构搜索方法，构造了一种融合自注意力、空间全连接层与空间池化的搜索空间，并通过权重共享技术显著降低了神经架构搜索的时间开销。本文利用长短时记忆网络生成网络架构，并通过基于策略梯度的 REINFORCE 算法训练智能体，以解决神经架构搜索中的黑盒优化问题。本文的神经架构搜索方法可以在单张显卡上利用约 3 小时的时间搜索到可用的神经架构，该架构在Flowers-102 图像分类数据集上取得了超越MetaFormer的分类准确率。

关键词： 神经架构搜索；视觉 Transformer；强化学习；策略梯度；图像识别

中图法分类号 TP301.6 算法理论

1 引言

Transformer[1]是一种基于自注意力机制的神经网络架构，对于词元 (token) 序列具有良好的特征提取能力。利用 Transformer 在大规模数据集上进行训练，再向下游任务进行迁移，是近几年自然语言处理的典型范式[2-5]。

Transformer在自然语言处理任务上的成功吸引了计算机视觉领域的关注。自注意力机制在计算机视觉任务上最早被用于增强卷积神经网络 (CNN) 的全局特征提取能力[6]。随着视觉 Transformer (ViT) 的提出[7]，Transformer 在计算机视觉领域引起了广泛的影响，用于解决视觉问题的 Transformer 模型不断推陈出新，在图像分类、目标检测、语义分割等任务中均取得了超越 CNN 的性能[8]。

然而，Transformer 在计算机视觉任务中有效性的根源并非是自注意力机制的应用。近期大量基于多层感知机 (MLP) 的骨干网络在计算机视觉领域取得了有竞争力的结果[9-11]，甚至将 Transformer 中的自注意力模块替换为全卷积也能达到较好的模型效果[12]。MetaFormer[13]的成功进一步证明了 Transformer 在视觉领域起效的关键在于其图像切片-Transformer 编码器堆叠的整体架构，编码器中对 token 进行混合的模块可以自由变化。

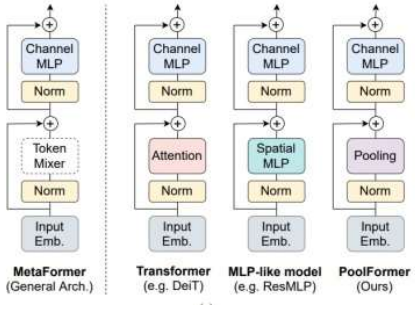

图 1 MetaFormer 中多样化的 token 混合模块

自注意力、空间 MLP、空间池化均可在视觉 Transformer 的基础架构中作为有效的 token 混合模块[13]。显然，如果使用某种特定的方式对三类 token 混合模块进行融合，那么有可能在视觉任务上取得超越单一 token 混合模块的结果。手工设计三种模块的融合形式需要大量的实验，且难以自动适应不同的视觉任务。神经架构搜索 (NAS) 是一种自动设计神经网络架构的技术，通过将神经网络设计视为高代价黑盒优化问题，可以用强化学习方法在特定的搜索空间上构造最佳的神经网络[14]。

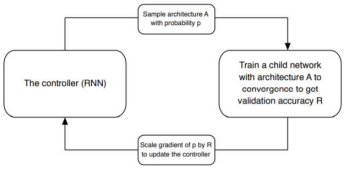

图 2 利用强化学习方法解决神经架构搜索问题

为了得到超越 MetaFormer 的有效视觉骨干网络，本文提出了一种基于 MetaFormer 的神经架构搜索算法 MetaFormer-NAS (MetaFormer-Neural Architecture Search)。本文主要的贡献有如下几点：

(1) 基于 MetaFormer 的基础结构，本文提出了一种融合三类 token 混合模块的新型视觉骨干网络搜索空间，利用长短时记忆网络 (LSTM) 作为网络生成控制器，并通过基于策略梯度的 REINFORCE 算法训练这一智能体。这是目前首个基于 MetaFormer 设计搜索空间进行神经架构搜索的工作。

(2) 为了降低智能体训练的代价，本文利用参数共享技术[15]实现了高效的神经架构搜索，使得架构搜索可以在单张显卡上利用约 3 小时时间完成。

(3) 本文在 Flowers-102 图像识别数据集测试了 MetaFormer-NAS 算法的识别精度，通过实验证明了这一神经架构搜索算法的有效性。

2 相关工作

2.1 视觉 Transformer 及其变体

本文的工作基于近年来对视觉 Transformer 有效性的研究。Transformer 是一种最初在自然语言处理任务中广泛应用的架构，适用于机器翻译和英语成分句法分析等[1]。在 Transformer 模型被提出以后，在未标注的文本上进行大规模预训练的新式 Transformer——BERT，在 11 项不同的自然语言处理任务上达到最先进水平[2]。

受到自然语言处理学界的启发，计算机视觉领域的研究者也将 Transformer 类似架构引入到这一 CNN 曾占主导性地位的领域中。在基于 Transformer 的视觉通用骨干网络中，ViT 是出现较早的模型，一经出现就达到了与卷积神经网络近似甚至超过的性能。ViT 将图片切分为大小相同的图片块，然后将这些块进行图片块嵌入，最后在图片块嵌入向量序列上应用多个 Transformer 编码器级联的网络架构进行特征提取[7]。微软亚洲研究院提出的 Swin Transformer，将卷积神经网络中的感受野与层次化思想融入视觉 Transformer 模型中，这一基于可调整窗口的模型刷新了多项视觉任务的性能指标[16]。

然而，近期出现了大量在视觉 Transformer 中替换自注意力模块的工作，这些视觉 Transformer 的非自注意力变体展现出了值得关注的性能。ConvMixer[12]将视觉 Transformer 中的自注意力模块等效的替换成卷积操作，取得了具有竞争力的结果。MLP-Mixer[9]提出了一种纯 MLP 的架构，仅依赖空间和通道 MLP，和 Transformer 的基本结构，就可以达到与 ViT 不相上下的结果。ResMLP[10]提出了一种与 Transformer 基本结构类似的纯 MLP 架构，可以有效的解决视觉 Transformer 收敛慢，缺乏数据效率 (data efficiency) 的问题。颜水成团队提出了 MetaFormer[13]，总结了视觉 Transformer 及其各类非自注意力变体的共性结构，并证明了空间池化可作为有效的 token 混合模块。本文基于 MetaFormer 的实验结果，构造了一种融合多种 token 混合模块的神经架构搜索空间。

2.2 基于策略梯度的强化学习

本文的神经架构搜索算法利用 LSTM 作为网络生成控制器，在神经架构搜索空间中进行探索，以生成性能最佳的网络为目标。为了训练这一智能体，本文采取了基于策略梯度的强化学习算法。

策略梯度算法在连续空间中的 RL 问题有广泛的应用，其主要思想是将策略π参数化表示化为π_θ，并计算出关于动作的策略梯度，然后沿着梯度的方向不断地调整动作，逐渐得到最优策略。REINFORCE 算法[18]是一种蒙特卡洛思想的应用，每次随机选择一个动作，并利用当前情节 (episode) 下环境的反馈调整参数。本文采用的 REINFORCE with baseline 方法，在 REINFORCE 算法的梯度更新公式中加入了基准项，从而有效降低梯度的方差、提升收敛速度。近端策略优化 (PPO) [19]也是经典的策略梯度方法，通过将在线学习转化为离线学习，可以有效的减少与环境互动的次数以提升学习效率。

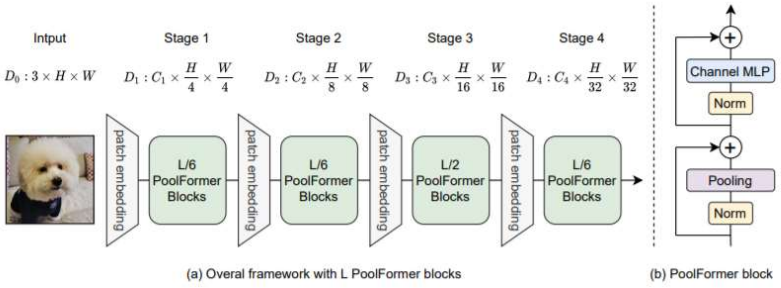

图 3 MetaFormer 的整体架构

2.3 基于强化学习的神经架构搜索

神经架构搜索是一种在特定的神经网络搜索空间中，利用黑盒优化算法寻找最佳的网络结构的技术。本文利用了神经架构搜索技术在 MetaFormer 的基础上对 token 混合模块的架构进行了探索。

强化学习和演化算法是解决黑盒优化问题的典型方法，本文主要利用强化学习解决神经架构搜索中的优化问题。NAS 和 RL 的结合最早可以追溯到 2017 年，谷歌大脑团队提出可以利用循环神经网络 (RNN) 作为控制器产生神经网络编码，然后训练对应的网络结构，以验证集精度作为奖赏反馈给控制器，并用策略梯度方法进行控制器训练[20]。

然而，神经架构搜索问题是一个典型的高代价黑盒优化问题，进行一次神经网络训练-评估的开销很大，必须设计合理的搜索策略来提升架构搜索的效率。NASNet[21]提出的如下的两点搜索策略：可以在小数据集上搜索架构，再将搜索到的网络迁移到大数据集上；可以搜索特定的重复单元，而不是搜索整个网络。

对于神经架构搜索与强化学习的结合，从优化目标上进行创新也是一种有效的思路。MNastNet[22]提出可以将单一准确率目标转化为多目标，同时考虑准确率和特定硬件上的时延，从而得到适用于特定硬件环境的神经架构。E2GAN[23]利用生成对抗网络 (GAN) 中生成器每一层均可产生图片的性质，创新性的将神经架构搜索建模为多阶段的马尔科夫决策过程而非简单的多臂老虎机问题。本文的神经架构搜索算法主要基于高效神经架构搜索 (ENAS) [24]实现，该算法使用了一种参数共享策略，并将优化目标设定为验证集上一个批次的准确率，进一步降低了搜索需要的时间。

3 方法描述

3.1 搜索空间构建

在 NAS 问题中，任何最终生成的计算图可以被视为一个超图的子图。与 ENAS[24]类似，我们将网络的搜索空间视为一个有向无环图 (DAG) 的子图，其中所有的节点表示无参数运算，而所有的边表示带有可训练参数的运算，并指明数据的流向。边上的可训练参数可以在不同的子图之间共享，从而有效的减少了网络训练的次数。

本文的神经架构搜索空间基于 MetaFormer 实现，下面首先说明空间中模型的基础架构。设 MetaFormer 的图像输入为I，MetaFormer 中首先进行图像切片与 token 嵌入，这一过程可表示为：

$$X = InputEmb(I) \qquad (1)$$

在模型早期阶段添加卷积对于提升 Transformer 性能的作用已经在部分工作中得到证实[26-27]。为了解决视觉 Transformer 在小数据集上难以收敛的问题，与 MetaFormer 不同，本文在图像切片与 token 嵌入之前引入了一层可学习的卷积层：

$$X = EarlyConv(InputEmb(I)) \qquad (2)$$

嵌入后的 token 被送入一串堆叠的 MetaFormer 编码器中，每个编码器均包括 token 混合模块、正则化模块与通道 MLP，计算过程如下所示：

$$Y = TokenMixer\big(Norm(X)\big) + X \quad (3)$$

$$Z = Channel\_MLP\big(Norm(Y)\big) + Y \quad (4)$$

本文采取了和 MetaFormer 类似的架构，将模型分为 4 个阶段，每个部分对特征图逐渐进行下采样，并包括不同数量的 MetaFormer 编码器。对于本文搜索空间中的模型，模型内的每一个 token 混合模块结构相同，每个模型间的差异只存在于 token 混合模块中。

在 MetaFormer 中设计融合自注意力、空间池化和空间 MLP 的 token 混合模块，我们参考 ENAS 中对于 RNN 单元的设计方式，使用了一种有N个节点的 DAG，其边表示共享参数的空间 MLP，而点表示无参数的恒等或空间池化操作。空间 MLP 的考虑到自注意力机制的参数量较大，我们不将其列入节点内部，而是在 token 混合模块的入口处将其融合进模型的计算图中。此处自注意力机制的实现细节与 Bottleneck Transformer[25]类似，将每一个空间位置上所有通道内的信息视为一个 Transformer token，并利用基于三角函数的绝对位置编码来融合位置信息。

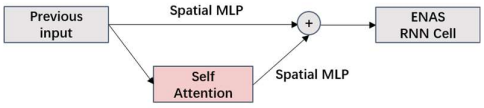

图 4 在 token 混合模块的入口处融合自注意力机制

在产生 token 混合模块的过程中，LSTM 控制器随机采样N个不同的决策。这里我们用一个$N = 4$情形下的一个具体的网络推导过程为例，来解释我们神经架构搜索的具体机制。

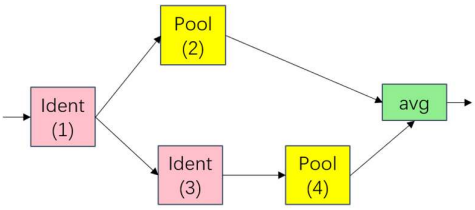

图 5 一种可能的 4 节点的 token 混合模块

用二元组$(prev_node, op)$表示 LSTM 做出的具体决策，其中$prev_node$表示 DAG 中节点的前驱，而op表示 DAG 中节点的无参数操作。假设 LSTM 控制器做出的 4 个决策按顺序分别为：*(0, Identity), (1, Pooling), (1, Identity), (3, Pooling)*。那么根据这一决策序列推导神经网络的具体过程可描述如下：

(1) 1 号节点以图 3 中描述的 token 混合模块入口模块作为输入，其$prev_node$无意义，根据当前节点 LSTM 生成的操作op（恒等），设 token 混合模块的输出为x，则本节点对应的输出为$k_1 = xW_1$。

(2) 2 号节点的前驱结点$prev_node$为 1 号节点，LSTM 生成的操作op为“池化”，则本节点对应的输出为$k_2 = Pooling(k_1 W_{1,2})$。

(3) 3 号节点的前驱结点$prev_node$为 1 号节点，LSTM 生成的操作op为“恒等”，则本节点对应的输出为$k_3 = k_1 W_{1,3}$。

(4) 4 号节点的前驱结点$prev_node$为 3 号节点，LSTM 生成的操作op为“池化”，则本节点对应的输出为$k_4 = Pooling(k_3 W_{3,4})$。

(5) token 混合模块只可能有一个输出，但是当前 DAG 中有 2 个出度为 0 的节点 (2 号、4 号)，对于这两个节点的输出计算均值，得到模块最终的输出$z = Average(k_2, k_4)$。模块最终的输出与残差连接相加后将被送往通道 MLP 中，进行公式(4)中的运算。

我们可以发现，任何两个节点之间均存在一个独立的权重矩阵，如前文的例子所示，控制器通过选择节点的前驱节点来选择使用哪一个权重矩阵，而搜索空间中的所有模型均共享相同的权重矩阵集合。

3.2 智能体训练与应用

生成神经网络的控制器是一个具有 64 个节点的 LSTM，这一控制器可以被视为在神经架构空间中进行探索的智能体。与 ENAS 类似，这一智能体通过 Softmax 分类器进行决策，模型的输入为空，每一步决策将被作为输入送入下一层中。

设控制器参数为θ，搜索空间中模型的共享参数为ω，我们的神经架构搜索过程由如下两个基本步骤组成：

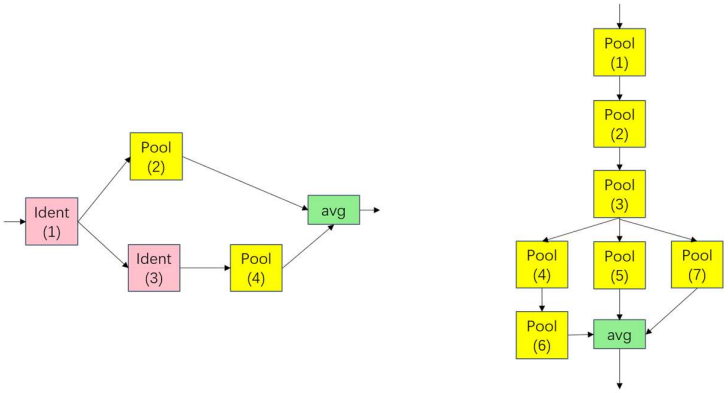

图 6 搜索到的 token 混合模块结构（4 节点、7 节点）

(1) 固定控制器参数θ，即固定控制器的策略$\pi(m;\theta)$，利用训练集中数据以梯度下降方法训练模型共享参数ω。考虑控制器产生神经架构时的随机性，要最小化的目标为$E_{m\sim\pi}[L(m;\omega)]$。此处的目标$L(m;\omega)$为共享模型在验证集上的交叉熵损失，而$m$指的是根据控制器的策略$\pi(m;\theta)$产生的模型。共享参数$\omega$的梯度可以用蒙特卡洛方法估计：

$$\nabla_{\omega}E_{m\sim\pi(m;\theta)}[L(m;\omega)] \approx \frac{1}{M}\sum_{i=1}^{M}\nabla_{\omega}L(m_i;\omega) \quad (5)$$

实验结果[24]证明，此处的蒙特卡洛估计在$M=1$的情况下有效，故我们只需要令控制器随机产生一个具体的模型，然后利用梯度下降法训练这一模型的权重ω即可。

(2) 固定模型参数ω，令控制器在搜索空间中进行多次探索，更新控制器参数θ，最大化奖赏$E_{m\sim\pi(m;\theta)}[R(m;\omega)]$，其中$R$为模型在验证集上一个批次的准确率。具体实践中，本文使用带有基线的 REINFORCE 算法对控制器参数进行训练。

轮流训练模型共享参数ω与控制器参数θ，周而复始，经过一定的轮次，即可得到能够生成有效 MetaFormer 架构的智能体，随后可以利用这一智能体产生有效的神经网络架构。由于控制器生成网络时带有随机性，在实践中一般进行多次随机采样，生成多种不同的架构，并从中选取一个奖赏函数R值最大的架构。这一架构将被应用于目标数据集，其参数将会重新初始化 (与搜索阶段的共享参数ω无关)，并完成训练-测试的模型验证流程。

4 实验验证

为了验证本文架构在图像分类问题上的有效性，本文在牛津大学 Flowers-102 数据集上进行了实验。图像大小统一为 224×224，并通过随机裁剪和随机水平翻转的方式进行了数据增强。

对比实验中，本文的模型基本与 PoolFormer-s12 具有相同的整体架构。不同之处主要有如下几点：

(1) 在模型早期加入了卷积以促进模型的收敛；

(2) 图像切片过程中的 token 数量被调整为 16×16 以减少自注意力模块和空间 MLP 模块的计算量；

(3) token 混合模块从单一空间池化改为通过神经架构搜索获得的新型 token 混合模块。

所有的实验均在一台安装了 Ubuntu 18.04 LTS 的服务器上进行，CPU 的型号为 Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz，GPU 的型号为 RTX Quadro 8000，显存大小约为 47.5G，CUDA 的版本为 11.4.

论文相关的源代码已在 github 仓库 **https://github.com/QwQ2000/MetaFormer-NAS** 上公开。

4.1 搜索时间

由于每个节点有两种可选的操作，本文提出的模型搜索空间理论上具有$2^N \times (N-1)!$种不

同的模型配置。本文在$N=4$和$N=7$两种情况下评测了神经架构搜索所需的时间开销，可以发现本文提出的方法可以在计算资源受限的情况下，以较短的时间产生有意义的神经架构。

节点数	时间	可选架构数
4	2.8h	96
7	3.1h	92,160

表 1 神经架构搜索所用的时间

进行神经架构搜索时，共进行了 100 个轮次，每训练一次共享参数令智能体探索 100 次。对共享参数进行训练时，令智能体随机产生一个确定的网络结构，并在整个训练集上以梯度下降法训练一个轮次。共享参数以 AdamW 优化器进行优化，学习率为 1e-3，权重衰减为 1e-4；控制器以 Adam 优化器进行优化，学习率为 3.5e-4。共享参数训练与奖赏函数值计算时所用的批次大小均为 128。

4.2 精度对比

经过约 3 个小时左右的搜索，MetaFormer-NAS 算法得到了如图 6 所示的神经架构。为验证本文算法的有效性，我们与 MetaFormer 的三种基本配置 (自注意力、空间 MLP、空间池化) 进行了对比。用于对比的 MetaFormer 均包括了早期阶段卷积和 token 数量调整。

网络架构	Flowers-102 准确率
MetaFormer-s12 + Pooling	64.9
MetaFormer-s12 + SpatialMLP	65.1
MetaFormer-s12 + MHSA	64.7
MetaFormer-s12 + 4 Node NAS	**64.5**
MetaFormer-s12 + 7 Node NAS	**65.9**

表 2 对比实验

所有模型均在 Flowers-102 数据集上训练了 600 个轮次，每经过 10 个轮次保存一次模型权重，最终用验证集精度最高的模型进行测试。模型训练使用 AdamW 优化器，学习率为 1e-3，权重衰减为 1e-4。观察结果，可以发现本文方法得到的 7 节点架构具有超过 MetaFormer 基本配置的准确率，这证明了 MetaFormer-NAS 算法的有效性。

结束语

本文提出了一种基于 MetaFormer 的新型神经架构搜索方法——MetaFormer-NAS，探究了利用神经架构搜索方法改进 MetaFormer 中 token 混合模块的可能性。本文利用基于策略梯度的强化学习方法和参数共享技术，在有限的计算资源下以极短时间搜索到了有意义的神经架构，并在 Flowers-102 数据集上进行了测试。本文的神经架构搜索技术具有通用性，可适用于多种不同的图像识别任务，对于医学影像等小数据集下的图像识别任务应当具有更好的结果。

本文的改进方向主要有以下几点：(1)由于受到计算资源和作业完成时间的限制，本文并未对模型进行充分调优，智能体的训练时间也极为不足，后续可对模型进行进一步调优，提升模型的精度；(2)本文对智能体进行训练时使用的是较为经典的 REINFORCE with baseline 算法，后续可以使用 PPO 等更为前沿的强化学习算法进行训练；(3)本文的搜索空间参数量较大，对于达成准确率-计算量平衡较为不利，后续可对搜索空间进行进一步改进，得到参数量较小的搜索空间，并以多目标优化方法得到准确率和推理速度兼备的高效网络架构。

致 谢

张嘉淏（第一作者）在创意提出、核心代码、实验数据处理与论文撰写上做出了主要贡献;刘添羽（共同作者）在强化学习算法方向对核心代码与论文撰写提供了帮助。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems. 2017: 5998-6008.

[2] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[3] Yang Z, Dai Z, Yang Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding[J]. Advances in

neural information processing systems, 2019, 32.

[4] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv preprint arXiv:1907.11692, 2019.

[5] Lan Z, Chen M, Goodman S, et al. Albert: A lite bert for self-supervised learning of language representations[J]. arXiv preprint arXiv:1909.11942, 2019.

[6] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.

[7] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[8] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[J]. arXiv preprint arXiv:2103.14030, 2021.

[9] Tolstikhin I, Houlsby N, Kolesnikov A, et al. Mlp-mixer: An all-mlp architecture for vision[J]. arXiv preprint arXiv:2105.01601, 2021.

[10] Touvron H, Bojanowski P, Caron M, et al. Resmlp: Feedforward networks for image classification with data-efficient training[J]. arXiv preprint arXiv:2105.03404, 2021.

[11] Ding X, Xia C, Zhang X, et al. Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition[J]. arXiv preprint arXiv:2105.01883, 2021.

[12] Anonymous, 2022. “Patches are all you need?”. In Submitted to The Tenth International Conference on Learning Representations (ICLR). under review.

[13] Yu W, Luo M, Zhou P, et al. MetaFormer is Actually What You Need for Vision[J]. arXiv preprint arXiv:2111.11418, 2021.

[14] Zoph B, Le Q V. Neural architecture search with reinforcement learning[J]. arXiv preprint arXiv:1611.01578, 2016.

[15] Pham H, Guan M, Zoph B, et al. Efficient neural architecture search via parameters sharing[C]. International Conference on Machine Learning. PMLR, 2018: 4095-4104.

[16] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[J]. arXiv preprint arXiv:2103.14030, 2021.

[17] 刘建伟,高峰,罗雄麟.基于值函数和策略梯度的深度强化学习综述[J].计算机学报,2019,42(06):1406-1438.

[18] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine learning, 1992, 8(3): 229-256.

[19] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[J]. arXiv preprint arXiv:1707.06347, 2017.

[20] Zoph B, Le Q V. Neural architecture search with reinforcement learning[J]. arXiv preprint arXiv:1611.01578, 2016.

[21] Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8697-8710.

[22] Tan M, Chen B, Pang R, et al. Mnasnet: Platform-aware neural architecture search for mobile[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 2820-2828.

[23] Tian Y, Wang Q, Huang Z, et al. Off-policy reinforcement learning for efficient and effective GAN architecture search[C]. European Conference on Computer Vision. Springer, Cham, 2020: 175-192.

[24] Pham H, Guan M, Zoph B, et al. Efficient neural architecture search via parameters sharing[C]. International Conference on Machine Learning. PMLR, 2018: 4095-4104.

[25] Srinivas A, Lin T Y, Parmar N, et al. Bottleneck transformers for visual recognition[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 16519-16529.

[26] Xiao T, Dollar P, Singh M, et al. Early convolutions help transformers see better[J]. Advances in Neural Information Processing Systems, 2021, 34.

[27] Hassani A, Walton S, Shah N, et al. Escaping the big data paradigm with compact transformers[J]. arXiv preprint arXiv:2104.05704, 2021.