



密级： 无

编号：  

# 国防科技大学

## 博士研究生学位论文 开题报告

论文题目： 基于启发式算法和可微分算法的  
神经网络架构搜索

学 号： 18023074 姓名： 胡凯

一级学科： 集成电路与工程

研究方向： 人工智能加速

指导教师： 罗莉, 王蕾 职称： 研究员, 副研究员

学 院： 计算机学院

开题时间： 2020 年 2 月 23 日

国防科技大学研究生院制

二〇一八年一月

# 说 明

- 一、开题报告应按下述要求打印后装订成册：
  1. 使用 A4 白纸，双面打印；
  2. 封面中填写内容使用小 3 号仿宋字体；
  3. 表中填写内容使用 5 号楷体字体。
- 二、封面中的编号由学院填写，采用八位数编码，前四位为审批日期，精确到年月即可，第五位为院别，后三位为审批流水号（按年）。如 15016100，为 6 院 15 年审批的第 100 位开题申请者，审批时间为 15 年 1 月份。院别代码与编制序列一致，海洋科学与工程研究院代码为 0。
- 三、开题报告表中学员填写的内容包括学位论文选题的立论依据、文献综述、研究内容、研究条件、学位论文工作计划、主要参考文献等，指导教师认可学员开题报告内容后，对学员学位论文选题价值、对国内/外研究现状的了解情况、研究内容、研究方案等方面予以评价。
- 四、开题报告评议小组一般由 3-5 名具有正高级专业技术职务的专家（包括导师）组成，其中一名为跨一级学科的专家。
- 五、博士生开题报告会应面向全校公开举行，评议小组听取研究生的口头报告，并对报告内容进行评议审查。
- 六、若开题报告获得通过，应根据评议小组意见对开题报告进行修改，并在开题报告会两周后，将评议表和修订后开题报告纸质版原件交学院备案；若开题报告未获得通过，则填报延期开题申请，由原开题报告评议小组重新组织开题报告会。

## 1. 学位论文选题的立论依据

### 1. 选题依据和研究意义

20 世纪 50 年代, 随着“感知器(Perceptron)”的出现, Frank Rosenblatt 发明了第一个神经网络原型。但是直到最近几年, 随着需要解决的任务变得越来越复杂, 简单的感知器模型已经无法满足现实需求, 对于深度网络的探索开始成为研究领域的热点。许多工作为深度网络的应用做出了重大的贡献, 各种各样的神经网络模型也在学术界和工业界被广泛应用。虽然神经网络在处理实际问题时, 展示出了强大的性能优势, 但是神经网络本身却是非常复杂的。

首先, 神经网络存在着多种类, 有卷积神经网络、循环神经网络、脉冲神经网络等等。每种神经网络在处理不同任务时, 具有自己独特的优势。具体地, 我们列举了 3 种常见的神经网络, 也是我在之后的研究中将会应用到的神经网络。

(1) 卷积神经网络(Convolutional Neural Networks, CNN)是仿照人类视觉机制构建的深度神经网络。在计算机视觉领域, 对于卷积神经网络的研究非常多。而且卷积神经网络对于图像和视频的处理也已经日臻成熟。目前, 许多人类专家设计的卷积神经网络性能非常优秀, 如 AlexNet、VGGNet、GoogLeNet 和 ResNet[1]。

(2) 循环神经网络(Recurrent Neural Network, RNN)是一类以序列数据为输入, 具有记忆性的一类深度神经网络, 例如长短期记忆网络(Long Short-Term Memory networks, LSTM)就是一种常见的循环神经网络。循环神经网络在自然语言处理(Natural Language Processing, NLP), 例如语音识别、语言建模、机器翻译等领域有自己的独特优势。


(3) 脉冲神经网络(Spiking Neuron Networks, SNN)被誉为是第三代人工神经网络, 它对大脑神经元的模拟则更加接近。液体状态机(Liquid State Machines, LSM)就是一种脉冲神经网络。其中的每个神经元是一个累积记忆单元, 当更新神经元的时候, 其值并不是邻近神经元的累加, 而是它自身的累加。一旦达到阈值, 它会将它的能量传递到其他神经元。

另外, 每一种类型的神经网络的结构组成又是非常复杂的。例如: 卷积神经网络的层数, 卷积层和池化层的排布; 循环神经网络的神经元排布, 以及激活函数的选择; 液体状态机网络中的神经元的连接以及激活神经元和抑制神经元的比例等等。这些网络的众多的不确定性造成了专家在设计一个具有良好性能的神经网络时面临着巨大的网络架构候选集合的挑战, 即网络架构的搜索空间过大, 不可能进行遍历。因此, 针对某个任务设计一个好的神经网络是一件很难的事情, 这需要人类专家具有很多的设计经验, 消耗大量的设计时间, 不断进行尝试、改进。另外, 设计网络架构除了专家的丰富设计经验外, 可能还需要一些运气。

最后, 对于深度神经网络来说, 大量数据的训练是至关重要的, 通过训练神经网络才能学习到数以百万计的参数。目前的神经网络的训练工作需要专用的计算加速器如 GPU(Graphic Processing Unit)和 Google 开发的 TPU(Tensor Processing Unit)的助力。虽然现代的硬件加速器越来越成熟, 但训练的过程仍旧是非常漫长的。验证网络架构的优劣是一件时间周期特别长的事情, 只有一些类似谷歌、亚马逊这些拥有大量的计算资源的大公司才有能力去做。

根据上面的论述, 在神经架构搜索(Neural architecture Search, NAS)出现之前, 想要设计出一个好的神经网络架构是一个相当耗费人力资源、计算资源和时间资源的工作。而且由于人类专家精力的限制, 往往不可能遍历很多的不同的网络架构, 这造成找到的网络的性能是次优的。不仅如此, 网络架构只能针对特定的任务来展开设计, 经过复杂的、长期设计的网络架构往往只能满足一种需求, 而在工业界对于不同任务的设计需求相当之

大。这就造成了网络架构设计的市场过大和网络架构设计人员过少的矛盾。

神经架构搜索在这样的背景下应用而生。神经架构搜索是 AutoML [2] 的一个子领域。AutoML 的概念最初源自 2012 年学术界提出一个新观念——rogramming by Optimization (PbO)，字面意思是指最优化的程序开发，实质上为解决编程时人工调校参数的问题，即学习构建其他机器学习算法的机器学习算法。简单来说，神经架构搜索就是将网络架构的设计工作完全交给机器来做，利用机器不断进步的强大算力来搜索出针对不同任务的、理想的神经网络架构，最终实现解放人类设计专家的密集的劳动力。

自动化神经网络搜索方法通过对神经网络架构的自动化设计，得到了学术界和工业界的广泛关注。在 2016 年，谷歌公司的学者们发表了论文并最先提出了这一概念，并且取得了很好的结果。自动搜索得到的神经网络架构在几个小型的经典数据集上击败同时代同级别的手工设计网络。另外，神经架构搜索也在工业界崭露头角。在 2017 年，亚马逊公司发布其神经架构自动化设计产品 Amazon SageMaker，旨在帮助开发人员和数据科学家快速轻松地构建、训练和部署任何规模的机器学习模型。用 AI 来设计 AI，让更多对机器学习了解有限的人，把 AI 技术运用到产品设计中，从而降低了使用机器学习的门槛，谷歌公司于 2018 年发布产品 Cloud AutoML Vision，微软同年也发布了它的 AutoML 工具，并将其加入了 Azure 云服务平台。在 2019 年，小米人工智能团队以及华为诺亚方舟团队也对神经架构搜索做了很多研究。

虽然神经架构搜索将架构设计过程自动化，但是神经架构搜索也面临着很多的问题。其中最大的问题就是对于计算资源的需求巨大，例如原始的 NAS 论文 [3] 为了得到最终的神经网络架构消耗了很长的运行时间（450 个 GPU，3-4 天）。这样大的计算规模下，在实验室的计算条件下很难进行，只有在一些独角兽网络公司才能进行。所以，如何在神经架构搜索中降低计算资源和时间资源的消耗将会是非常有意义的工作。

另外，神经架构搜索随着目前算力越来越大应用前景和应用场景也会越来越广。但是，神经架构搜索在类脑计算领域的工作还比较少，我希望在此次课题期间，将神经架构搜索和类脑计算这两个重要的领域联系起来，将由于算力增加带来的优势带给类脑计算。液体状态机是类脑计算中一个重要的网络类型，在有非常好的硬件可完成性、节能等优势。而针对液体状态机架构优化的研究甚少，大多网络架构都是人类专家手工设计的。将神经架构搜索的思想应用到液体状态机的架构优化上无疑能够设计出具有更好性能的网络。在未来，类脑领域得益于神经架构搜索思想的加持，将会得到更好的发展。

## 2. 文献综述

（该领域在国内/外的研究现状及发展动态；阅读文献的范围以及查阅手段等。博士不得少于 2000 字，硕士不少于 1000 字。可附页）

### 一、神经架构搜索 (Neural Architecture Search, NAS)

NAS 被视为 AutoML 的子领域，神经架构搜索刚刚兴起却发展极为迅速，相关的研究和文章也非常多。目前，神经架构搜索主要研究内容可以被切分为三个部分：搜索空间的定义、搜索策略的选择和架构评估策略的设计。如图 1 所示。首先，将不同的网络候选操作和各个操作之间的连接方式的所有的可能性都列出来构成一个巨大的搜索空间。然后根据不同的搜索策略来不断从搜索空间中选出更好的神经网络架构。最后，我们会将前面选出的架构根据特定的评估策略来进行评估，得到候选架构的性能度量，并且这个度量指标将会作为信号来更新搜索策略。这个过程是由机器自动化进行的，直至搜索出最终的架构。

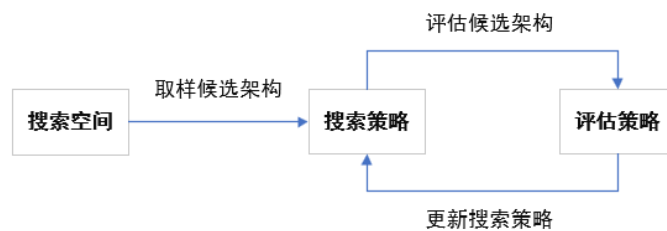


图 1 神经架构搜索的过程

下面从搜索空间的定义、搜索策略的选择和架构评估策略的设计这三个方面展开叙述：

#### 1. 搜索空间：

神经架构搜索空间包括一系列操作，比如：搜索卷积神经网络时搜索空间包括卷积层、池化层的链接次序，以及滤波器的大小、个数、步幅等超参数的选择；而在搜索循环神经网络时，搜索空间包括各个神经元节点的连接方式和激活函数（tanh, ReLU, identity, sigmoid）的不同选择。在确定了搜索空间包含的内容之后，定义一个搜索空间还要有诸多的权衡，下面三种类型的搜索空间定义是目前出现过，分别是链式架构搜索空间、多分支架构搜索空间和基于块而构建的搜索空间。其中基于块构建的架构搜索空间最为常见。

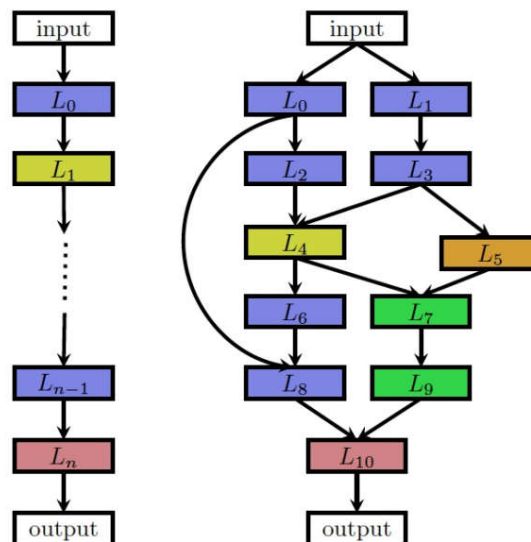


图 2 链式架构空间（左图）与多分支架构空间（右图）

原始的 NAS 论文搜索空间是简单的链式结构空间，顾名思义，就是搜索空间内的结构按照链的形式进行排列，如图 2 中的左图所示。链式架构空间的每一层输出都是下一层的输入。链式架构空间的好处是结构简单，便于理解，搜索空间规模不大。但是，它的缺点同样十分明显，就是网络深度有限而且网络架构单一。因为 VGG 19[4] 等模型已经证明直筒的链式结构容易造成梯度弥散，卷积网络无法使层数更深。所以，在 NAS 领域，链式搜索空间应用较少。

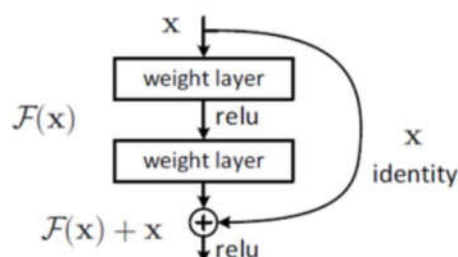


图 3 残差块架构

目前，现代的神经网络架构为了获得更好的性能，网络架构的的复杂性也逐步提升，例如 2016 年 Kaiming He [5] 凭借提出的深度残差网络不仅横扫了图像识别领域的各大比赛，而且还获得了 CVPR 最佳论文。深度残差网络解决了在深度网络在继续增加深度时会出现的在误差反向传递过程中梯度消失的问题。深度残差网络由多个如图 3 所示的残差块组成，残差块在输出时新增了跳跃连接 (Skip Connections) 而通过对传统层叠加的平原网络 (Plain) 新增跳跃连接，从而实现残差网络。在深度神经网络架构搜索的过程中，跳跃连接也是神经架构搜索中的重要候选操作之一。如图 2 右图中的网络架构就包含了多分支以及跳跃连接等连接方式。多分支架构空间颠覆了传统的链式架构空间的单一结构，为实现复杂的、表达能力更强大的网络提供了很好的帮助。如图 2 的右图所示，在链式架构空间的结构中，每个计算功能块的输出可以为多个不同的计算功能块提供输入，甚至可以实现跨层。这样复杂的网络架构空间容量巨大，其中的网络多变，网络的性能也更强，在最近的实验中，经常被应用。但是，巨大的益处也伴随着一定的缺陷。多分支架构空间的应用使得搜索空间呈指数级别增加。当要搜索的网络规模比较大时，多分支架构空间需要的计算资源是难以估量的。

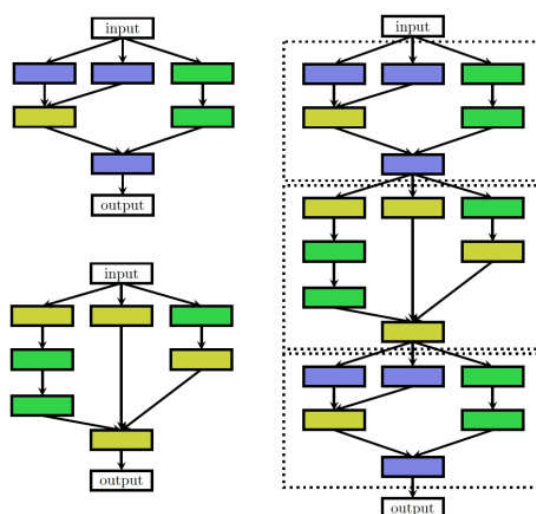


图 4 基于块构建的搜索空间

所以,目前常见的一种搜索空间结构是基于块构建的搜索空间。这是一种对于链式架构搜索空间、多分支架构搜索空间的一种折中的处理方式。块内使用多分支架构搜索空间,而在块间使用链式架构搜索空间。具体做法是:将在多分支架构搜索空间上搜索得到的基本块通过改变堆叠来生成新的网络架构,一方面可以减少优化变量数目,另一方面得到的基本块可以在不同任务之间进行迁移。Zoph 团队[2]就在多分支架构搜索空间上找到了两种基本块: normal cell 和 reduction cell。之后随这两种基本块进行排列组合生成一个性能很好的深度为 21 层的卷积网络架构,具体组合的方式是,堆叠 normal cell 6 次之后连接一个 reduction cell,再重复这种组合方式 3 次,得到了完整的卷积网络结构。虽然简单地堆叠两种基本块得到的架构性能是次优的,但是要得到很深的网络架构,这种折中处理是降低计算开销的一种很好的方式。

## 2. 搜索策略

搜索算法通常是一个迭代过程,定义了使用怎样的算法可以快速、准确找到最优的网络结构参数配置。常见的搜索方法包括:随机搜索、贝叶斯优化、进化算法、强化学习、基于梯度的算法等等。其中强化学习和进化学习是主流算法。

基于强化学习的神经架构搜索方法的核心思想是通过一个 RNN (或者是 LSTM [35]) 控制器网络在搜索空间中得到一个子网络架构 (child network),然后用这个子网络架构在数据集上训练,在验证集上测试得到准确率,再将这个准确率作为信号 (signal) 回传给控制器网络,控制器网络根据得到的信号继续优化得到另一个子网络架构,如此反复进行直到得到最佳的结果,即收敛。整个过程称为神经架构搜索。谷歌大脑的许多 NAS 工作 [2], [6], [7], [8], [9] 都是采用这种强化学习 [10] 的方式来处理的,并且取得了好的结果。比如原始的 NAS 在 CIFAR-10 数据集上取得的错误率为 2.89%,超过了当时的手工设计神经网络的性能。

另外一个主流的搜索策略是进化学习 (Evolutionary algorithms) [11], [12], [13]。为了达到自动搜索高性能的神经网络结构的目的,需要进化一个模型种群 (population)。每一个子模型,也就是种群中的每个个体 (individual) 都是一个训练过的架构。模型在验证集 (validation dataset) 上的准确度就是度量个体质量或适应性的指标。在一次进化过程中,随机从模型种群中选出两个个体;根据优胜劣汰对模型进行识别,不合适的模型会立刻从模型簇中被剔除,即代表该模型在此次进化中的消亡;而更优的模型则成为母体 (parent model),通过交叉变异进行繁殖得到新的子模型;子模型被创造出来之后,经过训练并在验证集上对其性能进行评估之后,把优秀的子模型放回到模型种群中。此时,该子代则成为母体继续进行上述几个步骤的进化。Yukang Chen [14] 等人的团队提出了增强进化神经结构搜索 (RENAS),这是一种具有 NAS 强化突变的进化方法,将强化变异整合到神经架构搜索的进化算法中,其中一个变异控制器用于学习轻微修改的影响并进行变异动作。最终,在 CIFAR-10 数据集上取得了竞争优势。进一步迁移到 ImageNet 数据集时,该网络实现了最先进的精度。

另外,可微分方法 DARTS 在 2018 年被 Liu 等人提出。传统的用遗传算法和 RL 的 NAS 使用的是离散的、不可微的搜索空间,而这篇文章将架构松弛为连续的搜索空间,可以使用梯度下降法对验证集性能进行优化。在 CIFAR-10, ImageNet, Penn Treebank [15], WikiText-2 数据集上的实验表明取得了先进的性能。作者认为强化学习和进化算法方法效率低下的原因是:将架构搜索看作是离散空间下的黑盒问题,因此产生了大量的架构评估需求,而 DARTS [16] 是在连续的搜索空间下直接使用误差反向传递。另外, DARTS 不需要控制器网络,并且有良好的可迁移性。如式 (1) 所示,把选择候选操作这个离散的事件,通过 softmax 函数将候选操作事件连续化,而 softmax [18] 函数的导数很好求得。点  $(i, j)$  的操作被参数化为一个维度为  $|O|$  的  $\alpha(i, j)$  向量,就可以通过概率值判断出  $i, j$  边



上应该选择哪种操作。其中两个节点之间无连接也设计为一种操作，即 zero 操作。

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x) \quad (1)$$

虽然 DARTS 连续化搜索空间，使得搜索任务可以使用可微分的方式进行误差反向传递，使得对神经架构搜索的理论理解更加清晰，但是原始的 DARTS 也存在着一些缺点。候选操作的垄断性偏置是其最大的缺点，即在架构搜索的早期，某种操作的先天优势会被很大程度地放大，导致架构搜索很容易收敛到只有一种操作占据垄断地位的架构，例如跳跃连接和 identity 激活函数就很容易在架构搜索中取得垄断优势。为了解决这一缺陷，小米人工智能实验室提出了 Fair DARTS [17]，用合作竞争方法来解决这个问题。具体做法是：给每个操作一个独立的架构权值来缓解垄断性优势，添加一个辅助的损失函数来最小化候选操作先天优势带来的偏置性。

除了强化算法、进化算法和可微分算法之外，一些启发式算法也作为搜索策略被应用到神经网络架构搜索当中。粒子群优化算法 (Particle Swarm optimization, PSO) [19]、和声优化算法 (Harmony search algorithm) [20] 和模拟退火算法 (Simulated annealing algorithm) 等算法也被用到了神经架构搜索领域。目前，禁忌算法 (Tabu Algorithm) [21] 还没有被使用在神经架构搜索中，但是作为一种高效的搜索算法，对于提高搜索的效率是非常有意义的。但是，禁忌算法也存在缺陷，就是很容易陷入局部最优解，可以考虑加入更多的随机性因素来改善这一情况。

另外，随机搜索 (Random Search) 在最近得到了很多的关注。瑞士联邦理工学院的一个团队 [22] 提出了一个很有争议的观点：平均而言，随机策略优于最先进的 NAS 算法。作者认为目前先进的 NAS 算法得到的候选架构的性能排名不能反映候选架构的真实排名；另外，虽然显著地提升了搜索效率，但是广泛使用权值共享会对架构的训练产生负面影响，从而降低了搜索过程的有效性。作者认为当搜索的架构数量很大时，随机搜索的策略是有更好搜索性能的搜索策略。这也让学术界更加重视随机搜索的重要性。

NAS 领域的搜索策略还有很多种[28], [31]，每一种策略对于处理不同的任务都有自己独特的优势。在选择搜索策略的过程中，需要根据目标的不同合理选择并改进[29], [30], [32]。

### 3. 评估策略：

性能评估是指评估候选架构性能的过程。最直接的方式是对候选架构执行标准训练和验证过程。但训练每个子网络架构需要巨大的资源消耗，这样会间接地限制了可以探索的网络架构的数量。因此，选择近似度量方式来衡量架构的性能是目前神经架构搜索领域内的主流做法。虽然近似估计会影响搜索性能，但是会大幅度减少搜索时间，提高搜索效率。目前，常见的近似评估方法有以下几种：将每个子模型只训练几代，而不是训练到收敛；另外，权值共享是一种显著减少性能评估时间的非常聪明的方式。

权值共享思想来源于迁移学习。传统的迁移学习往往是在数据集层面上的迁移学习，但是网络架构上的权值也是可以进行迁移学习的，这样可以发挥这些权值的经验性能。作者认为将一个架构训练到收敛，但这样只是为了得到精度作为强化学习的奖励，然后就丢掉了所有训练好的权值参数是非常浪费计算资源的做法。ENAS 的贡献是让所有的子网络架构共享一个提前训练好的样本的权值，避免了从零开始训练，大大减少训练的时间。值得一提的是，如式 (2) 所示，权值可以共享的基础是每次只取样一个样本，即  $M = 1$ 。论文并没有提到这一做法的理论原理，但是在实验中，表明这样做是可以工作的。



$$\nabla_{\omega} \mathbb{E}_{\mathbf{m} \sim \pi(\mathbf{m}; \theta)} [\mathcal{L}(\mathbf{m}; \omega)] \approx \frac{1}{M} \sum_{i=1}^M \nabla_{\omega} \mathcal{L}(\mathbf{m}_i, \omega) \quad (2)$$

最终，ENAS 在 CIFAR-10 数据集上取得了 2.89% 的错误率，和原始的 NAS 2.65% 的错误率相当；在 PTB (Penn Treebank) 数据集上取得了 55.8 的困惑度，这是当时的顶尖性能。ENAS 在取得很好的性能的同时，最大的进步是在于搜索效率的提升，相比于原始的 NAS，有 1000 倍的搜索加速比。

谷歌大脑团队[23]的另一个工作将共享权值的思想发挥得更彻底。作者认为网络架构生来不平等，在一些任务中某些架构就是表现更好。所以，作者想放大网络架构拓扑在性能上的优势，缩小网络超参数的影响，即探究在不学习任何权重参数的情况下，单凭网络结构能在多大程度上对给定任务提供解决方案。作者在评估阶段，对于所有的候选网络架构，使用从均匀随机分布中采样的单个共享权重参数来填充全部连接，并测量预期性能。这样，比 ENAS 更有侵略性缩减了搜索时间，即共享权值也不必训练，而是直接给定一个数字。

另外，也有很多工作是训练代理模型来对候选架构的性能进行预测，从而省略了候选架构训练和评估过程。谷歌大脑团队提出的 PNAS [6] 是采用逐渐进阶的方式来训练性能预测器，但是最终的预测期的性能并不理想。华为诺亚方舟团队也有类似的方式，训练代理预测器来评估候选架构性能。加入性能预测器加速评估阶段也是很有前途的一种近似评估的方式。

## 二、液体状态机

液体状态机是一种脉冲神经网络[27]。上文对神经架构搜索的针对对象都是卷积神经网络和循环神经网络，在做了一些文献研究后，发现 NAS 也可以应用到搜索液体状态机(LSM)网络当中。不仅如此，液体状态机网络还具有其独特的优势。在使用 NAS 搜索 RNN 网络时，需要消耗大量的训练时间。相比于 RNN，LSM 的结构具有先天的优势，LSM 的训练代价会显著减少。因为 LSM 的组件液体池 (Liquid) 中神经元权值是随机分布的，不需要进行训练，只需要训练读出层 (Readout) 层，这会大大减少训练代价。其次，LSM 的结构是硬件友好的，便于硬件实现，相较于其它的复杂网络结构更容易制造和进入市场。另外，LSM 是仿脑、异步实现的新一代网络，可以处理 DVS (Dynamic Vision Sensor) [24] 这种基于事件的数据，所以在硬件上实现也是更加节能的。

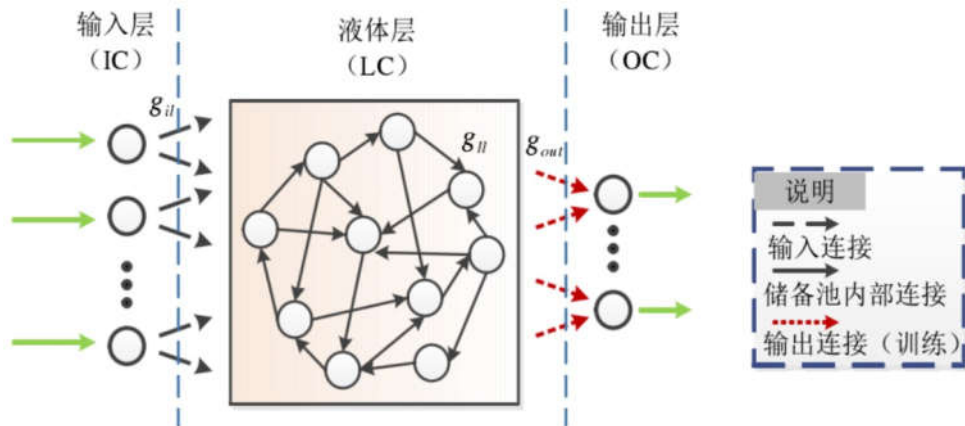


图 5 LSM 的结构图[25]

下面将简单介绍液体状态机网络的原理：

首先，LSM 主要分成三个部分：输入层 (Input)、液体层 (Liquid) 和输出层 (Readout)。如图 5 所示，输入层可以是模拟输入也可以是离散的脉冲输入，输入神经元和网络层所有的神经元相连。液体层是 LSM 的网络层，网络由脉冲神经元递归连接组成，突触连接的方向和权值大小都是随机的。输出层由输出神经元组成，主要是输出网络的结果，输出层的神经元和网络层中的而每个神经元都相连。液体状态机的特点在于它能够有效地处理时变的时间序列，并且能够实时地输出结果。

液体状态机网络由于它简单的网络结构，在最近得到了很多的应用，但是对于液体状态机的网络架构的研究却不是很多。虽然液体状态机存在着很强的随机性，但是网络架构对于其性能的影响也是很大的。如果将液体 (Liquid) 内部的各种神经元之间的比例、连接概率、连通度等参数进行优化，那么将会很大程度上提升 LSM 的性能。这些都还是停留在网络超参数优化的领域。

2019 年，耶鲁大学的一篇论文 [26] 对 LSM 的大液体池进行集群分块，通过集群的连接调整，提高了语音、图像的分类精度，同时也减少了内存需求。这涉及到了 Liquid 中小集群的架构拓扑的概念。这可以使用神经架构搜索的思想来搜索获得性能理想的液体状态机网络。具体分析之后，发现相对于强化学习、进化算法、各种启发式算法等搜索策略，可微分算法对于 LSM 网络架构搜索的适应性更好。首先，可微分算法在连续的搜索空间下，可以实现 Liquid 中的小集群之间的串并联关系，使得连接更具多样性。另外，相比于其他的 NAS 搜索算法，可微分算法能够提供 Liquid 中的小集群之间回环连接的可能。

### 3. 研究内容

#### 3.1 研究目标

- (1) 采用启发式算法作为搜索策略来解决 RNN 网络架构的搜索问题,并取得性能和搜索时间之间的平衡;
- (2) 采用启发式算法进一步搜索 CNN 网络架构,并取得性能和搜索时间之间的平衡;
- (3) 采用可微分算法作为 NAS 搜索策略,来搜索 LSM 网络架构,并获得领先性能;

#### 3.2 主要研究内容及拟解决的关键科学问题和技术问题

##### 3.2.1 主要研究内容:

###### (1) 使用禁忌算法搜索 RNN 网络架构

首先, 缩减搜索空间, 使得搜索更有效率; 其次, 通过增加全局解的随机性来改进的禁忌算法, 使得采用这种改进的搜索策略能够更好地平衡对搜索空间中的解的全局探索和局部开发, 提高搜索质量; 另外采用权值共享技术来加速整个架构搜索的过程。最后, 在 PTB 和 Wiki-txt2 数据集上搜索出具有顶尖性能的 RNN 网络架构。

###### (2) 使用启发式算法搜索 CNN 架构

首先, 根据算力和一些先验知识确定搜索空间; 其次根据具体问题选取搜索效果最好的启发式算法作为搜索策略; 在候选架构的性能评估上使用权值共享思想来加速搜索过程。最后, 能在 CIFAR-10 和 ImageNet 数据集上搜索到具有领先性能的 CNN 网络。

###### (3) 使用可微分方法搜索 LSM 网络架构

首先根据 LSM 网络的具体特点, 定义好搜索空间; 其次, 将搜索空间进行连续化操作; 然后, 将需要传递的超网架构误差进行适当的形式化定义; 之后, 利用可微分算法对于 LSM 网络架构进行搜索, 直到收敛。使得整个算法能够在 Minst、nMinst、Caltech-101 [33] 和 FSDD 等数据集上取得顶尖性能, 另外还可以在 DVS 数据集上取得良好地性能。

##### 3.2.2 拟解决的关键科学问题:

###### (1) 在用 DARTS 算法搜索 LSM 架构时, 会有下面 3 个问题:

- 1) 解决 DARTS 在反向传递的梯度信息的形式化表示问题, 使得架构搜索中的损失函数能够正确引导架构搜索向着更好性能的方向进行;
- 2) 神经架构搜索中的搜索空间的定义问题很关键, 因为良好的搜索空间能使得搜索更加高效。特别是在 LSM 架构搜索中, 可以借鉴的工作非常有限, 定义好搜索空间是亟待解决的关键性问题。
- 3) 对原始的 DARTS 论文进行初步实验和分析后发现, 在 DARTS 寻找最优架构时会出现偏置现象, 即在架构搜索的早期, 某种操作的先天优势会被极端放大, 导致架构搜索很容易收敛到只有一种操作占据垄断地位的架构, 例如跳跃连接和 identity 激活函数就很容易在架构搜索中取得垄断优势。

##### 3.2.3 拟解决的关键技术问题:

- 1) 液体状态机网络在 Brian2 模拟器中的实现方式, 以及如何快速得到性能指标;
- 2) NAS 的计算 (主要是评估成本) 如何在多 GPU 的情况下并行执行。

### 3.3 拟采取的研究方法、技术路线、实施方案及可行性分析

#### 3.3.1 技术路线

本课题分为三个研究内容，分别是利用启发式搜索算法，搜索循环神经架构和卷积神经架构，在保证搜索出的架构性能的前提下，提高搜索效率。另外一个研究内容是可微分的 NAS 搜索算法来搜索液体状态机网络架构。下面将针对这三个研究内容分别给出具体

##### (1) 使用禁忌算法搜索 RNN 网络架构

首先，使用禁忌算法等启发式算法作为 NAS 的搜索策略，但是原始的禁忌算法是存在缺点的，在搜索过程中很容易陷入局部最优解当中。如算法 1 所示，可以考虑在搜索过程中加入更多的随机性，使得获得解的质量更高。具体来说，每次对当前解的领域探索  $S$  步之后，会引入一个全局随机解。这样做的好处是能够更好地平衡全局解的探索和局部解的开发。

```

1: Initialize:  $tabu\_list \leftarrow \emptyset, tabu\_size, R, S$ 
2: for  $round = 1 \rightarrow R$  do
3:   if  $round == 1$  then
4:      $model.arch = INITIALARCHGENERATOR()$ 
5:      $tabu\_list.append(model.arch)$ 
6:   else
7:      $model.arch = RANDOMSAMPLEARCH()$ 
8:   end if
9:   for  $step = 1 \rightarrow S$  do
10:     $new\_model.arch =$ 
11:       $NEWARCHGENERATOR(model.arch)$ 
12:    if  $new\_model.arch \notin tabu\_list$  then
13:       $new\_model.ppl =$ 
14:         $ARCHEVALUATE(new\_model.arch)$ 
15:    end if
16:    while  $new\_model.ppl < Tbest$  do
17:      if  $len(tabu\_list) < tabu\_size$  then
18:         $tabu\_list.append(new\_model.arch)$ 
19:         $Tbest = new\_model.ppl$ 
20:      end if
21:    end while
22:     $model.arch = tabu\_list[-1]$ 
23:  end for
24: end for

```

算法 1 随机性增强的禁忌搜索算法

下一步工作是改进 RNN 架构搜索的搜索空间。在调研了一些其他团队工作之后发现，ENAS 和 DARTS 这两个工作在搜索 RNN 架构的时候，得到的最终架构中几乎只包括  $ReLU$  和  $\tanh$  这两个激活函数。而且通过查阅资料和分析之后发现其原因是，激活函数 sigmoid 在找到的 RNN 架构已经存在[34]，而激活函数 identity 在简单的、层数较少的 RNN 网络中只能提供很有限的非线性特性。所以，缩减搜索空间是一个更有效的手段。因为在缩减的搜索空间内，更容易获得性能较好的 RNN 网络架构，能够有效提高搜索效率。

最后，要在候选架构评估过程中引入权值共享技术。权值共享技术思想的来源为迁移学习，即一个架构的权值分布可以迁移到另一个架构上。目的是能够进一步减少搜索时间，提高搜索效率。另外，受到 Adam Gaier 工作的启发，可以考虑采取更具有侵略性的参数共享技术——只训练共享参数一次。这样做的有两个好处：①. 相比较于 ENAS，只训练一次共享参数可以大幅度提高搜索效率；②. 影响网络性能的因素主要是网络架构拓扑和网络权值参数，而通过简单训练得到的网络权值使得该网络的性能更大程度上依赖于网络架构带来的增益。通过这种方式，更容易获得先天就很好的网络架构拓扑。之后，再将得到的架构进行完整的训练，得到好的网络权值参数。这样最终的神经网络将会是具有更好性能的神经网络。

## (2) 使用启发式算法搜索 CNN 架构

在完成以改进的禁忌算法为搜索策略进行 RNN 网络架构之后,继续探索其它更适合 CNN 架构搜索的启发式算法来作为接下来搜索工作的搜索策略,目的是减少搜索时间,提高搜索效率。在这一研究内容下,技术路线如下:

首先,需要确定 CNN 网络架构的搜索空间。这里,计算点上可以施加的操作有 5 种,分别是:跳跃连接、卷积核为  $3 \times 3$  和  $5 \times 5$  的卷积操作、核大小为  $3 \times 3$  的最大池化和平均池化操作。初步确定搜索空间是基于块构建的,设计出两类基本块——Conv cell 和 Reduction cell,在将这两种基本块进行如图 6 式的堆叠,最后构成完整的神经网络架构。

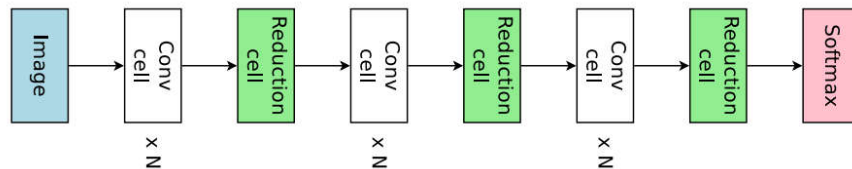


图 6 基于块构建的网络架构

在搜索空间中的每个候选架构都用有向无环图 (Directed Acyclic Graph, DAG) 来表示。如图 7 所示,假设整个 CNN 网络架构有 6 个计算点。计算点之间连接规则是:每个计算点只能和前节点连接(如节点 3 只能接受节点 1 和节点 2 的输出作为输入,而不能接受节点 4、节点 5 和节点 6 的输出作为输入)。每个计算点的不同颜色代表不同的操作。

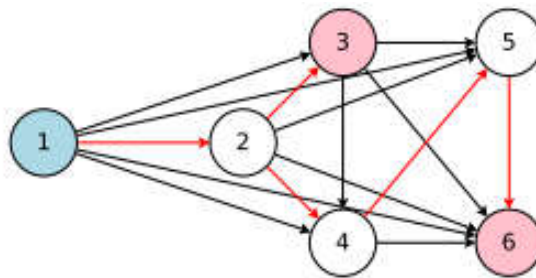


图 7 CNN 网络架构的搜索空间示意图

其次,是确定以何种启发式算法作为 NAS 的搜索策略。选择标准是在较短的搜索时间内能够搜索出在 CIFAR-10 和 ImageNet 数据集上具有顶尖精度的架构。在技术实现上,可以通过实验对比来进行选择。目前,候选的启发式算法包括模拟退火算法、粒子群算法两种。最后在候选架构的性能评估时,同样会使用权值共享技术来加速。

## (3) 使用可微分方法搜索 LSM 网络架构

首先,将 LSM 网络架构中的拓扑、候选操作等构建成为一个搜索空间。因为 LSM 中 Liquid 是基于概率分布的,而这些概率更容易被连续化。这也是选择可微分架构搜索的主要原因,可以说可微分架构搜索对于 LSM 的适应性更好。

之后可以将 LSM 网络架构的搜索空间通过 softmax 函数对架构操作进行放松——成为连续的搜索空间。最后将整个架构搜索问题简化成优化一个如图 8 所示矩阵的问题。下面简单介绍一下这一矩阵。矩阵每一行表示一个计算点 (Node),而总共构建几个计算点可以进行指定。指定的标准主要根据在实验中究竟几个计算点使得最终架构的性能更好,这里假设共有  $n$  个计算点。每个计算点由 3 中属性构成,分别是:①. 神经元占比,②. 计算点的固有属性,③. 与其他点的连接概率。神经元占比指的是,在 Liquid 中神经元数量一定的情况下,每个计算点将分得神经元的百分比;而计算点的固有属性包括:总神经元中兴奋神



经元占比、兴奋神经元到兴奋神经元连接概率 (E-E)、兴奋神经元到抑制神经元连接概率 (E-I)、抑制神经元到兴奋神经元连接概率 (I-E) 和抑制神经元到抑制神经元连接概率 (I-I); 与其他计算点的连接概率指的是本计算点和其它 (n-1) 个计算点的连接概率, 这一项会直接体现在 LSM 的网络拓扑上。

[[神经元占比, Node的固有属性, 与其他点的连接概率], # Node 1  
 [神经元占比, Node的固有属性, 与其他点的连接概率], # Node 2  
 [神经元占比, Node的固有属性, 与其他点的连接概率], # Node 3  
 .....  
 [神经元占比, Node的固有属性, 与其他点的连接概率]] # Node n

图 8 优化矩阵

然后采用梯度下降 (Gradient Descent) 的技术对优化矩阵进行不断优化, 直到收敛。在这一步需要首先确定传递的损失是每次优化后得到的优化矩阵所表示的候选架构在实际任务中的精度。这里, 需要采用 Brian2 模拟器来评估候选架构的精度。评估过程中, 还需要逐个训练每个候选 LSM 架构的 Readout 层, 这里 Readout 层采用的是单层全连接神经网络。

至此, 可微分 LSM 架构搜索的过程已经基本完成, 但是目前可微分架构搜索 DARTS 存在的偏置性缺陷需要解决。偏置性缺陷指的是在架构搜索的早期, 某种操作的先天优势会被极端放大, 导致架构搜索很容易收敛到只有一种操作占据垄断地位的架构, 例如跳跃连接和 identity 激活函数就很容易在架构搜索中取得垄断优势。这里, 采用 Fair DARTS [17] 中采用的合作竞争方法来解决。具体做法是:

①. 通过 (3) 式的放松方式来给每个操作一个独立的架构权值来缓解垄断性优势, 相比于 (1) 式, 所有的操作都能协同地进行优化, 有前途的操作还能继续增加权值来减少损失值, 直到被选;

$$\bar{o}_{i,j}(x) = \sum_{o \in \mathcal{O}} \sigma(\alpha_{o,i,j}) o(x). \quad (3)$$

②. 添加一个如式 (4) 所示的辅助损失函数 (zero-one loss) 来最小化候选操作先天优势带来的偏置性。这一函数的主要作用是将计算点之间的连接概率不断推向接近 0 或 1 的概率值。因为这些概率值一旦接近 0.5, 就更容易导致选择的不确定性。

$$L_{0-1} = -\frac{1}{N} \sum_i^N (\sigma(\alpha_i) - 0.5)^2 \quad (4)$$

在完成以上工作之后, 实验会在 Python+Brian2 模拟器的环境下进行, 实验的数据集包括 Minst、nMinst、Caltech-101、FSDD 等数据集以及 DVS 数据集。初步实验计划总共设置 1000 个神经元, 5 个计算点。

### 3.3.2 可行性分析

本课题的目标是在较短时间内, 能够自动化搜索出具有良好性能的神经网络架构。首先, 对于 RNN、CNN、和 LSM 网络的架构研究已经比较成熟, 架构对于网络性能的影响是至关重要的。在编程方面, 目前已经能初步完整实现搜索得到具有顶尖性能的循环神经网络。所以使用相似的方法也可以实现卷积神经网络架构的设计。另外, 可微分架构搜索



方法DARTS已经证明了利用可微分方法可以成功搜索到顶尖性能的循环神经网络和卷积神经网络架构。虽然液体状态机网络架构搜索还没有工作证明可行，但在准确定义液体状态机网络的搜索空间并得到损失函数的准确形式化表示之后，用可微分方法搜索液体状态机网络架构是具有可行性的。其次，实验室具有高性能算法支持，可以保证整个实验过程中的算力资源充足。另外，图书馆数字资源和实验室连接互联网的设备可以满足我对于信息和知识的获取。最后，我的导师有多年神经网络方面的研究经验，可以对我进行指导，师门的师兄和师姐们也可以给我很多帮助。

### 3.4 预期创新点



- 1) 使用改进的禁忌算法作为NAS的搜索策略，这是第一次禁忌算法被应用在RNN架构搜索的任务当中。其次，计划使用只包含两种激活函数的缩减的搜索空间。这两个举措对于搜索质量的提升有重要意义。另外，在候选架构的评估过程中，使用更加激进的参数共享技术（仅训练共享参数一次），目的是最大程度上放大架构拓扑对网络性能的影响。这样能够显著提升搜索速度，相较于ENAS的搜索效率，取得10-20倍的加速比。
- 2) 对比多个启发式算法（包括禁忌算法、模拟退火算法、粒子群算法），选择性能最优的算法作为NAS的搜索策略，在候选架构的性能评估阶段引入权值共享技术。在保证找到优秀性能的CNN网络架构的前提下，进一步降低搜索时间。
- 3) 使用可微分的方法来对一个新的网络——液体状态机网络架构进行搜索，使得搜索到的液体状态机能够在Minst、nMinst、Caltech-101、FSDD等数据集以及DVS数据集上取得网络性能的提升，之后还将对搜索过程继续进行优化以提升搜索的速度。

首先，这是第一次将NAS思想用于解决液体状态机网络的搜索任务。目前，几乎所有的NAS工作都是在搜索CNN和RNN网络。而选择搜索LSM架构的原因主要有以下三点：①. 相比于RNN，LSM的结构具有先天的节能优势，可以更容易部署到外部设备当中；②. 因为LSM的组件液体池（Liquid）中神经元权值是随机分布的，不需要进行训练，只需要训练读出层（Readout）层，这会大大减少训练代价；③. 另外，LSM是仿脑、异步实现的新一代网络，是非常有前景的一种网络。还可以处理DVS这种基于事件的数据，所以在硬件上实现也是更加节能的。

另外，使用NAS算法中的可微分神经架构搜索来搜索LSM架构是另一个创新点。因为相对于强化学习、进化算法、各种启发式算法等搜索策略，可微分算法对于LSM网络架构搜索的适应性更好。首先，可微分算法在连续的搜索空间下，可以实现Liquid中的小集群之间的串并联关系，使得连接更具多样性。另外，相比于其他的NAS搜索算法，可微分算法能够提供Liquid中的小集群之间回环连接的可能。

#### 4. 研究条件

开展研究应具备的条件及已具备的条件，可能遇到的困难与问题和解决措施。

已具备的条件：

1) 该课题的研究内容主要依赖于计算机软件的模拟，主要使用 Python 语言和 PyTorch 作为实现平台。另外，将神经架构搜索应用到液体状态网络架构的搜索时，采用 Brian2 作为模拟软件平台，来评估找到的液体状态机网络的性能。

2) 其次，实验的所需的硬件已经具备，有英伟达 GeForce 1070 Ti 显卡，能够很大程度上加速神经网络架构的性能评估过程。而且，实验的软件模拟环境已经搭建完毕。并且，已经进行了初步的探索性实验，能够使用随机性加强的禁忌算法，结合参数共享的近似评估策略在缩小的搜索空间上搜索得到一个具有顶尖性能的循环神经网络。在之后的实验当中，计划继续实现对卷积神经网络的架构搜索，这就需要更多的计算资源的支持。目前本人所在的课题组有提供高性能服务器的能力，即有 4 个 Tesla P100 显卡。所以未来的实验条件将有足够的计算资源保障。

3) 另外，老师和师兄具有相关领域的研究、实践基础。

可能遇到的问题：

1) 对于卷积神经网络搜索的经验不足；

2) 对于处理服务器中显卡的并行计算的经验不足；

3) 在对液体状态机网络架构的搜索过程中，对于 Brian2 模拟器的使用经验不足；

4) 在使用可微分方法进行液体状态机网络架构的搜索过程中，对于要进行反向传递的误差的表示存在疑问。

解决措施：

1) 阅读经典书籍和论文，总结和分析文献中的重要知识点，进行反复学习；

2) 尽快对服务器的使用熟悉起来，学会并行性编写代码。希望能够通过提高算力来加速整个实验的进展；

3) 加强对液体状态机网络的理解，并在 Brian2 模拟器上进行简单的实例应用；

4) 对于反向传递的梯度表示的疑问，之后还会与论文作者取得联系进行一些沟通；

5) 加强和老师、师兄师姐们的沟通，虚心请教，请求指导。

## 5. 学位论文工作计划

起讫日期	主要完成研究内容	预期成果
2020 年 03 月 01 日 —— 2020 年 04 月 01 日	使用禁忌算法完整实现循环神经网络架构搜索	完整实现 NAS，搜索得到顶尖性能的 RNN 网络
2020 年 4 月 01 日 —— 2020 年 5 月 01 日	对以上的工作进行搜索卷积神经网络架构的补充实验	得到顶尖性能的 CNN 网络
2020 年 5 月 01 日 —— 2020 年 6 月 01 日	调研液体状态机网络相关工作、学习并实践 Brian2 模拟器	做一些关于液体状态机的探索性实验
2020 年 6 月 01 日 —— 2020 年 9 月 01 日	使用 DARTS 算法对于液体状态机网络架构进行搜索	搜索到具有顶尖性能的 LSM 网络
2020 年 9 月 01 日 —— 2020 年 11 月 01 日	撰写毕业论文	毕业论文
2020 年 11 月 01 日 —— 2020 年 12 月 01 日	论文预审、论文修改、准备答辩	完成答辩

注：每个子阶段不得超过 3 个月；预期成果中必须包含成果的形式、数量、质量等可考性指标该计划将作为论文研究进展检查的依据。

6. 主要参考文献（博士不少于 50 篇、外文不少于 25 篇，硕士不少于 30 篇、外文不少于 15 篇，可附页）

序号	文献目录（作者、题目、刊物名、出版时间、页次）
1	Canziani A, Paszke A, Culurciello E. An analysis of deep neural network models for practical applications[J]. arXiv preprint arXiv:1605.07678, 2016.
2	F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Automatic Machine Learning: Methods, Systems, Challenges. Springer, 2018, in press, available at <a href="http://automl.org/book">http://automl.org/book</a> .
3	B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," arXiv preprint arXiv:1611.01578, 2016.
4	K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
5	Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In CVPR
6	C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 19-34.
7	H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean, "Efficient neural architecture search via parameter sharing," arXiv preprint arXiv:1802.03268, 2018.
8	M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2820 - 2828.
9	B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing Neural Network Architectures using Reinforcement Learning," arXiv:1611.02167 [cs], November 2016.
10	L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," Journal of artificial intelligence research, vol. 4, pp. 237-285, 1996.
11	E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 4780-4789.
12	Thomas Elsken, J Hendrik Metzen, and Frank Hutter. 1804. Efficient Multiobjective Neural Architecture Search via Lamarckian Evolution. ArXiv e-prints (1804).
13	M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in Proceedings of the Genetic and Evolutionary Computation Conference. ACM, 2017, pp. 497 - 504.

序号	文献目录（作者、题目、刊物名、出版时间、页次）
14	Chen Y, Meng G, Zhang Q, et al. Renas: Reinforced evolutionary neural architecture search[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 4787-4796.
15	M. Marcus, G. Kim, M. A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger, "The penn treebank: annotating predicate argument structure," in Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, 1994, pp. 114 - 119.
16	H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," arXiv preprint arXiv:1806.09055, 2018.
17	Chu X, Zhou T, Zhang B, et al. Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search[J]. arXiv preprint arXiv:1911.12126, 2019.
18	E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144, 2016
19	Y.-F. Huang and J.-S. Liu, "Optimizing convolutional neural network architecture using a self-adaptive harmony search algorithm." B. Wang, B. Xue, and M. Zhang, "Particle swarm optimisation for evolving deep neural networks for image classification by evolving and stacking transferable blocks," arXiv preprint arXiv:1907.12659, 2019.
20	Y.-F. Huang and J.-S. Liu, "Optimizing convolutional neural network architecture using a self-adaptive harmony search algorithm."
21	Glover F, Laguna M. Tabu search[M]//Handbook of combinatorial optimization. Springer, Boston, MA, 1998: 2093-2229.
22	C. Sciuto, K. Yu, M. Jaggi, C. Musat, and M. Salzmann, "Evaluating the search phase of neural architecture search," arXiv preprint arXiv:1902.08142, 2019.
23	A. Gaier and D. Ha, "Weight agnostic neural networks," 2019.
24	Leñero-Bardallo J A, Serrano-Gotarredona T, Linares-Barranco B. A 3.6 $\mu$ s Latency Asynchronous Frame-Free Event-Driven Dynamic-Vision-Sensor[J]. IEEE Journal of Solid-State Circuits, 2011, 46(6): 1443-1455.
25	杜梅, 赵怀慈. 手写数字识别的研究[J]. 计算机工程与设计, 2010(15):132-135.
26	Wijesinghe P, Srinivasan G, Panda P, et al. Analysis of liquid ensembles for enhancing the performance and accuracy of liquid state machines[J]. Frontiers in neuroscience, 2019, 13: 504.
27	M. C. Cardoso, M. Silva, M. M. B. R. Vellasco, and E. Cataldo, "Quantum-Inspired Features and Parameter Optimization of Spiking Neural Networks for a Case Study from Atmospheric," Procedia Computer Science, vol. 53, pp. 74 - 81, 2015.

序号	文献目录（作者、题目、刊物名、出版时间、页次）
28	K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, and E. P. Xing, “Neural Architecture Search with Bayesian Optimisation and Optimal Transport,” in Advances in Neural Information Processing Systems 31. Curran Associates, Inc., 2018, pp. 2020 – 2029.
29	H. Cai, J. Yang, W. Zhang, S. Han, and Y. Yu, “Path-Level Network Transformation for Efficient Architecture Search,” in Proceedings of the 35th International Conference on Machine Learning, 2018.
30	A. Brock, T. Lim, J. Ritchie, and N. Weston, “SMASH: One-shot model architecture search through hypernetworks,” in International Conference on Learning Representations, 2018.
31	H. Cai, T. Chen, W. Zhang, Y. Yu, and J. Wang, “Efficient architecture search by network transformation,” in AAAI Conference on Artificial Intelligence, 2018.
32	Jin-Dong Dong, An-Chieh Cheng, Da-Cheng Juan, Wei Wei, and Min Sun. 2018. PPP-Net: Platform-aware Progressive Search for Pareto-optimal Neural Architectures. In ICLR.
33	Putra I. Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101[D]. Institut Teknologi Sepuluh Nopember, 2016.
34	J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, “Recurrent highway networks,” in Proceedings of the 34th International Conference on Machine Learning–Volume 70. JMLR. org, 2017, pp. 4189 – 4198.
35	F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” 1999.



## 7. 指导教师对开题报告的评语

（对 1-6 项逐项予以评价，并着重对国内/外研究现状的了解情况、研究内容的创新性等方面进行评价，最终给出是否满足博士/硕士层次学位论文研究要求的综合评价意见）

导师签名：

年 月 日