

NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING

Barret Zoph*, Quoc V. Le

Google Brain

{barretzoph, qvl}@google.com

【强化学习 107】Neural Architecture Search



张楚珩

清华大学 交叉信息院博士在读

27 人赞同了该文章

Neural architecture search (NAS) 是强化学习的一个重要应用方向，也是 AutoML 的一个非常火的研究方向。这里主要结合我们组学弟 @五更琉璃 在组会上的两次报告来梳理一下该方向的主要研究进展。

原文传送门

NAS (ICLR 2017): [Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." arXiv preprint arXiv:1611.01578 \(2016\).](#)

NASNet (CVPR 2018): [Zoph, Barret, et al. "Learning transferable architectures for scalable image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.](#)

ENAS (ICML 2018): [Pham, Hieu, et al. "Efficient neural architecture search via parameter sharing." arXiv preprint arXiv:1802.03268 \(2018\).](#)

特色

个人感觉一个应用场景要适合强化学习解决有两点比较重要：首先，得是一个顺序决策问题；其次，得有一个可以做试验的环境。在这两点上 neural architecture search (NAS) 问题是比较适合用强化学习来解决的。这里主要介绍该领域几篇比较经典的工作，让大家对于强化学习的这一个重要应用方向有一个大致的了解。

过程

1. AutoML

AutoML 领域一个重要的方向就是神经网络结构搜索。目前比较主要的有两个分支：基于强化学习的方法（比如 NAS）和基于可微分神经网络结构搜索的方法（比如 DARTS）。从效果上来看，前者效果上较好；但是从计算资源上来看，前者一般需要较大规模的计算资源。

2. NAS

大致结构

NAS 的目标是找到一个合适的神经网络结构，用于在某个或者某类任务上有更好的泛化性能。如下图所示，这篇文章使用了一个 RNN 的控制器，用该控制器采样得到某一个神经网络结构 A，在

该神经网络结构下训练数据并且得到相应的验证集上的准确率 R，使用该准确率来表征本次搜索得到的神经网络结构的好坏，进而将此作为信号来训练 RNN 控制器。大致结构如下图所示。

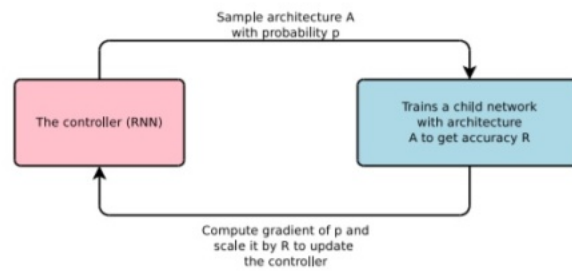


Figure 1: An overview of Neural Architecture Search. 知乎 @张楚珩

生成深度卷积神经网络结构

这里考虑生成一个深度卷积神经网络，RNN 控制器的每五步输出对应一层卷积神经网络；当生成神经网络的深度超过预先设置的值之后，就停止输出。接下来，用这里生成的神经网络结构去训练神经网络，并且记录该神经网络在验证集上的误差。利用该误差来更新 RNN 控制器的权重。

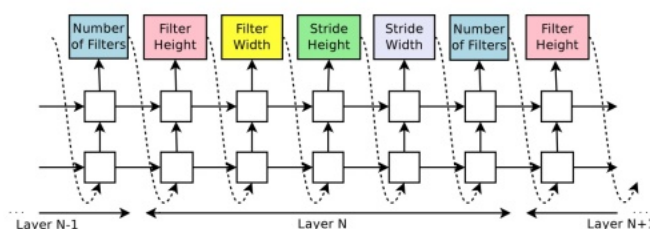


Figure 2: How our controller recurrent neural network samples a simple convolutional network. It predicts filter height, filter width, stride height, stride width, and number of filters for one layer and repeats. Every prediction is carried out by a softmax classifier and then fed into the next time step as input.

用强化学习来训练控制器

RNN 权重的更新方式如下：把 RNN 控制器中每一步吐出来的结果看作是强化学习中的行动，对应的状态就是控制器到第 t 步为止生成的网络结构。该强化学习问题在一个轨迹结束之后会给出一个 sparse reward/cost，也就是该神经网络结果对应的验证集上损失。这样就可以使用强化学习方法来更新 RNN 控制器权重。文章中使用了 REINFORCE with baseline：

$$\frac{1}{m} \sum_{k=1}^m \sum_{t=1}^T \nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) (R_k - b)$$

异步并行训练

强化学习中的一些策略梯度方法可以并行和异步地更新网络权重从而加速训练（比如 A3C）；这里也用了类似的方法来加速控制器的训练。如下图所示，每一个蓝色框相当于强化学习里面采样的一个轨迹，那么一个绿色框就对应了从一个 batch 中计算得到的梯度；为了能够异步进行多个绿色框，需要一些红色框来综合各个绿色框得到的梯度，这样绿色框下一轮可以从采样拿到最新的神经网络参数。

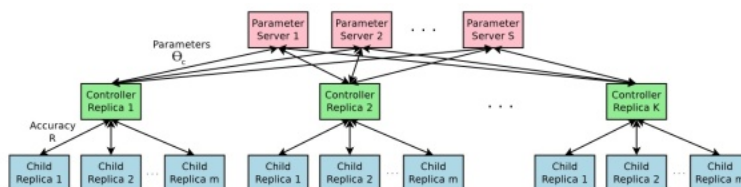


Figure 3: Distributed training for Neural Architecture Search. We use a set of S parameter servers to store and send parameters to K controller replicas. Each controller replica then samples m architectures and run the multiple child models in parallel. The accuracy of each child model is recorded to compute the gradients with respect to θ_c , which are then sent back to the parameter servers.

生成带有 skip connection 的卷积神经网络

接下来文章还设计了可以使得 RNN 控制器生成 skip connection 的方法，大致上就是把 RNN 的输出结构做了相应的规定。相应地，还可以扩展生成 learning rate、batchnorm、pooling 等结构。

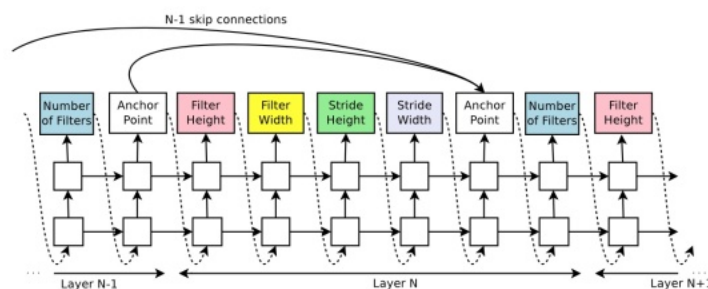


Figure 4: The controller uses anchor points, and set-selection attention to form skip connections.

生成深度 RNN 网络

每一步上 RNN 网络都可以看做输入是 h_{t-1}, a_t, q_{t-1} 输出是 h_t, q_t 的一个网络。这里事先规定了它们之间的连接关系，然后用 RNN 控制器去生成相应的激活函数和组合方式。具体可以参见下图，最左边

是认为规定的大致结构，中间是 RNN 控制器生成的结果，右边是 RNN 控制器生成结果对应的网络结构。

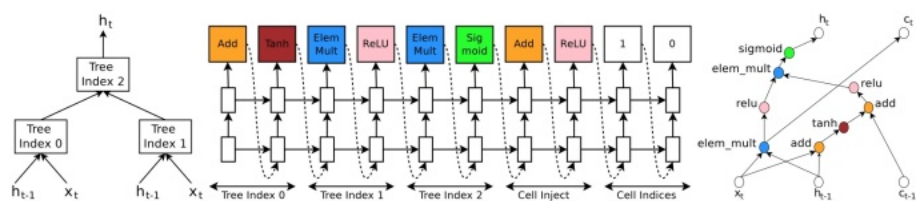


Figure 5: An example of a recurrent cell constructed from a tree that has two leaf nodes (base 2) and one internal node. Left: the tree that defines the computation steps to be predicted by controller. Center: an example set of predictions made by the controller for each computation step in the tree. Right: the computation graph of the recurrent cell constructed from example predictions of the controller.

缺点

- Limited AutoML : fixed layers, fixed number of nodes, fixed operation pools (conv, skip connect, batchnorm)
- Can not create operations like conv
- Too costly for only small datasets – CIFAR, Penn Treebank

3. NASNet

这篇工作的目标就是要搜索到一个好的神经网络结构，使它能够在较大的数据集（ImageNet）有好的效果。由于在较大的数据集上反复训练肯定是很费时间的，因此文章在这里就考虑先在小的数据集（CIFAR-10）上找神经网络结构，然后把它应用到大的数据集上。那么如何实现这个 transfer 呢？基于 sota 的一些网络结构设计，文章发现，很多好的网络结构都是由一些 cell 组成的，每个 cell 包含一组神经网络的 layers。因此，本文先人为设定神经网络的“宏观”结构，然后使用搜索的方法来找到有用的“微观”结构。

ps. 这篇文章用 NASNet 这个词指他们搜到的最好的那个网络结构，而不是这个方法。

文章设定神经网络的“宏观”结构如下：

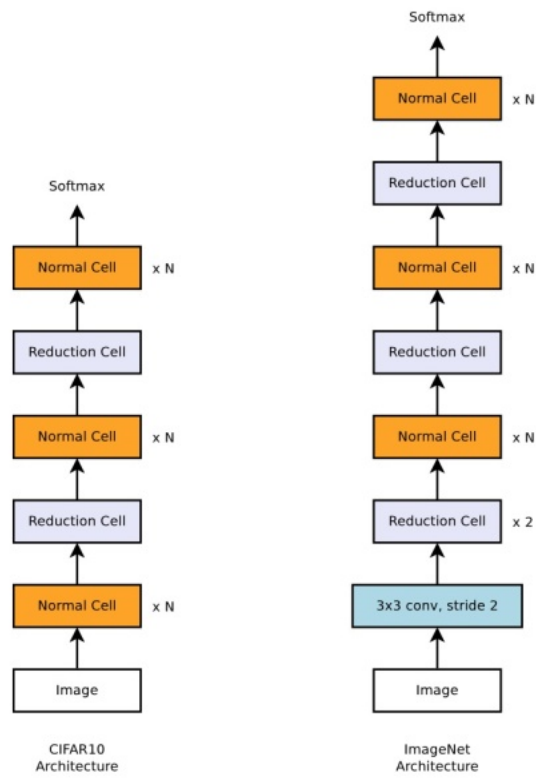


Figure 2. Scalable architectures for image classification consist of two repeated motifs termed *Normal Cell* and *Reduction Cell*. This diagram highlights the model architecture for CIFAR-10 and ImageNet. The choice for the number of times the Normal Cells that gets stacked between reduction cells, N , can vary in our experiments.

每一个 cell 含有多个 layers。其中 normal cell 的输入输出维度会保持不变，而 reduction cell 的长宽都会缩减一半。接下来，和前一个工作类似，文章使用 RNN 控制器来在小数据集上搜索合适的神经网络结构。RNN 和其所表达的 cell 结构如下图所示，一个 cell 由 B 个 block 组成，每个 block 如右图所示；RNN 从本 cell 的输入或者上一个 cell 的输入中，二选一，分别选择一个作为第一个输入或者第二个输入。然后接下来再分别指定这两个输入的变化方式和聚合方式。这样的搜索空间定义更加合理，可以生成 residual link 等结构。

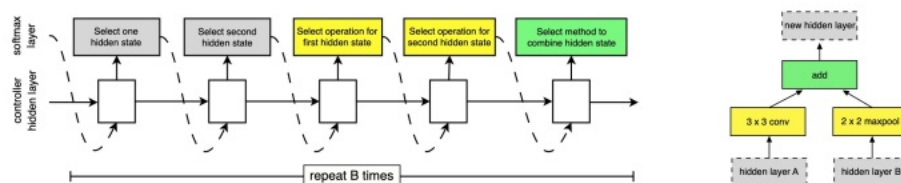


Figure 3. Controller model architecture for recursively constructing one block of a convolutional cell. Each block requires selecting 5 discrete parameters, each of which corresponds to the output of a softmax layer. Example constructed block shown on right. A convolutional cell contains B blocks, hence the controller contains $5B$ softmax layers for predicting the architecture of a cell. In our experiments, the number of blocks B is 5.

其搜索得到的结构如下图所示

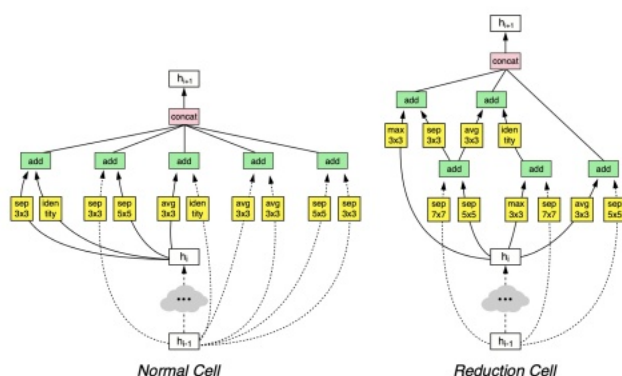


Figure 4. Architecture of the best convolutional cells (NASNet-A) with $B = 5$ blocks identified with CIFAR-10. The input (white) is the hidden state from previous activations (or input image). The output (pink) is the result of a concatenation operation across all resulting branches. Each convolutional cell is the result of B blocks. A single block corresponds to two primitive operations (yellow) and a combination operation (green). Note that colors correspond to operations in Figure 3.

最后，其实验效果也是非常明显的

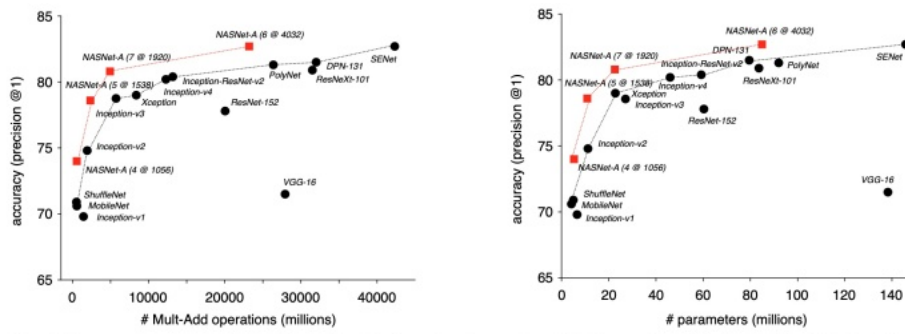


Figure 5. Accuracy versus computational demand (left) and number of parameters (right) across top performing published CNN architectures on ImageNet 2012 ILSVRC challenge prediction task. Computational demand is measured in the number of floating-point multiply-add operations to process a single image. Black circles indicate previously published results and red squares highlight our 125,000-scale models.

4. ENAS

ENAS 的全称是 efficiency neural architecture search。前一个方法虽然在小数据集上训练然后再迁移，但实际上仍然非常消耗计算资源，需要使用 450 个 GPU 训练三到四天。这里提出一种新的方法，来有效地搜索神经网络结构。

我们前面看到，神经网络结构搜索最消耗计算资源的地方在于：每次生成一个新的神经网络结构之后，为了评价这个神经网络结构的好坏，我们需要把这个神经网络在数据集上初始化权重，然后从头开始训练。这里就提出，神经网络的权重需要和控制器来同时学习，从而节省大量训练资源。这里面困难的地方就在于，神经网络的结构变化之后，可能对应的参数（甚至对应参数的 shape）都会发生变化。

神经网络结构规定为有向无环图

这里提出的解决方案就是，把整个神经网络结构用固定个节点（包含其中的神经网络参数）和相应的连边（包含相应的非线性函数种类）表示，RNN 控制器输出一个 DAG（有向无环图）和相应的连边性质，从而构成相应神经网络结构。对于一个 RNN 的结构搜索来说，结构和 RNN 控制器的输出对应关系如下图所示。

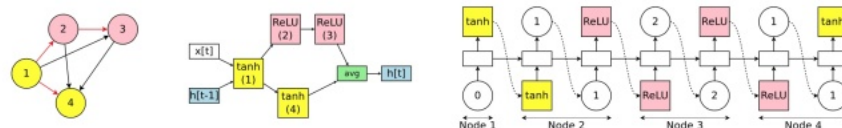


Figure 1. An example of a recurrent cell in our search space with 4 computational nodes. *Left*: The computational DAG that corresponds to the recurrent cell. The red edges represent the flow of information in the graph. *Middle*: The recurrent cell. *Right*: The outputs of the controller RNN that result in the cell in the middle and the DAG on the left. Note that nodes 3 and 4 are never sampled by the RNN, so their results are averaged and are treated as the cell's output.

即，每两步处理一个节点，先输出该节点和之前的哪个节点相连，然后输出相应的变化关系是什

么。最后该 DAG 中没有被用作输入的节点就全部自动综合起来得到最后的输出。

对于 CNN 来说，相应的对应关系也差不多。

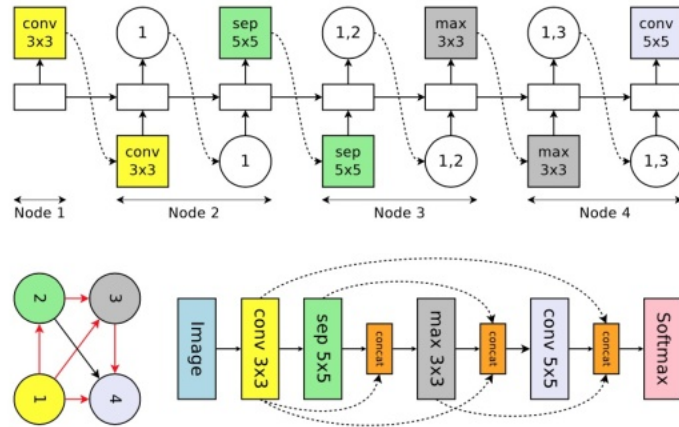


Figure 3. An example run of a recurrent cell in our search space with 4 computational nodes, which represent 4 layers in a convolutional network. *Top:* The output of the controller RNN. *Bottom Left:* The computational DAG corresponding to the network's architecture. Red arrows denote the active computational paths. *Bottom Right:* The complete network. Dotted arrows denote skip connections.

知乎 @张楚珩

文章后面还仿照 NESNet 的方法，人为设计“宏观”结构然后自动发现好的 cell 结构。

子模型权重学习和控制器学习同步进行

DAG 每一个节点被称作一个子模型。本文的重要贡献就是在学习控制器的同时学习子模型权重，这样不需要每次产生一个新的神经网络权重之后就重新训练一下整体的神经网络权重。每轮迭代分为两个步骤：训练各个子模型权重、训练控制器。

子模型权重的训练就是根据当前控制器，采样得到一个网络结构，利用该网络结构在数据集上的梯度，更新一步自己的权重

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

公式中, m 代表采样得到的神经网络结构, θ 为控制器权重, ω 为子模型权重, 相应的损失函数为当前神经网络结构在数据集上的损失函数。

接下来, 同时再利用 REINFORCE 来更新控制器参数。

Automatic recipe search

最近同学们因为疫情被关在家里面应该都厨艺大涨吧。我最近也在思考, 有没有人用机器学习方法来搜索一下食材之间的最优搭配和相应的制作方法呢? 我想, 食材的搭配和做法之间肯定是有一些隐含的科学道理的, 比如口感上的互补、味觉上的协调、营养上的均衡等。千百年来人们逐渐发现了一些百吃不厌的搭配, 比如"土豆+牛肉+烧", "土豆+青椒+鸡肉+黄焖", "冬瓜+排骨+炖"等。但是这并不是全部, 现在人们也在持续地发现一些好的"爆款"搭配, 从而获取巨大的商业利益, 比如乐乐茶、喜茶等发明的"草莓/葡萄+芝士+奶+茶"的搭配迅速走红, 促使其获得了 90 亿的估值 [1]。

而我不仅搞理论也搞搞实验, 最近也在家发现了一些比较不错的搭配。



(a) Spaghetti with Sichuan sausage and smoked meat



(b) Toast with fried egg, grilled pineapple and yoghurt



(c) Coco milk tea with chicken and Sichuan bean curd

[1] 钛媒体: 喜茶估值90亿, 值吗?

编辑于 2020-03-04

AutoML

深度学习 (Deep Learning)

强化学习 (Reinforcement Learning)

赞同 27

1 条评论

分享

喜欢

收藏

...

文章被以下专栏收录



强化学习前沿
读呀读paper

进入专栏