Bayesian Meta-network Architecture Learning

*Albert Shaw[1], *Bo Dai[1,2], Weiyang Liu[1], Le Song[1]

[1]Georgia Institute of Technology, [2]Google Brain

December 22, 2018

Abstract

For deep neural networks, the particular structure often plays a vital role in achieving state-of-the-art performances in many practical applications. However, existing architecture search methods can only learn the architecture for a single task at a time. In this paper, we first propose a Bayesian inference view of architecture learning and use this novel view to derive a variational inference method to learn the architecture of a *meta-network*, which will be shared across multiple tasks. To account for the task distribution in the posterior distribution of the architecture and its corresponding weights, we exploit the *optimization embedding* technique to design the parameterization of the posterior. Our method finds architectures which achieve state-of-the-art performance on the few-shot learning problem and demonstrates the advantages of meta-network learning for both architecture search and meta-learning.

1 Introduction

There has been much recent work focusing on designing novel structures for neural networks, *e.g.*, LeCun and Bengio (1995); He et al. (2016); Huang et al. (2016); Zhang et al. (2017). However, due to the combinatorial nature of the design space, hand-designing architectures is expensive and potentially sub-optimal. Developing techniques to automatically search this space has become a large focus of recent efforts, and several evolutionary and reinforcement learning based algorithms have been quite successful in achieving state-of-the-art performances on several tasks (Zoph and Le, 2017; Zoph et al., 2017; Real et al., 2018). Various methods have even seen some success in greatly reducing the search time (Liu et al., 2018; Pham et al., 2018; Cai et al., 2018b,a). However, even though methods use search spaces which are designed to be generalizable, the existing search methods are strongly task dependent and should be repeated for each new task. Meta-learning methods(Finn et al., 2017; Nichol and Schulman, 2018) however, allow the networks to be quickly trained on new data and new tasks. Since many base network structures, *e.g.*, VGG (Simonyan and Zisserman, 2014), Inception (Szegedy et al., 2016), and ResNet (He et al., 2016), can often be applied to different tasks with small variations, it is natural to ask whether we can train a meta-network to simultaneously learn a distribution of both weights and architectures which can be applied to many tasks with only slight modifications.

2 A Bayesian Inference View of Architecture Searching

In this section, we reformulate neural network architecture learning as Bayesian inference. This new view of neural network architecture learning inspires an efficient algorithm which can provide a task-specific neural network with adapted weights and architecture with only a *few* learning steps.

We can consider neural network architecture learning as an operation selection problem. Specifically, starting from the a DenseNet (Huang et al., 2017) architecture where the k-th layer of the neural network is defined as

**Both authors equally contributed to the paper.

$$x_k = \sum_{i=1}^{k-1} \left(z_{i,k}^\top \mathcal{A}_i \left(\theta \right) \right) \circ x_i := \sum_{i=1}^{k-1} \sum_{l=1}^{L} z_{i,k}^l \phi_i^l \left(x_i; \theta \right),$$

where $\mathcal{A}_i \left(\theta \right) = \left[\phi_i^l \left(\cdot; \theta \right) \right]_{l=1}^L$ denotes a group of operations from $\mathbb{R}^d \to \mathbb{R}^p$ depending on the parameters θ and $z_{i,k} \sim \mathcal{C}ategorial \left(\alpha_{i,k} \right)$ with $\alpha_{i,k}^l \geqslant 0$, $\sum_{l=1}^L \alpha_{i,k}^l = 1$. As we can see, such a neural network structure selects the operations on each of the outputs of the previous layers to form the output of the current layer. Assume the probabilistic model as

$$\begin{aligned} \theta &\sim \mathcal{N} \left(\mu, \sigma^2 \right), \\ z_{i,k} &\sim \mathcal{C}ategorial \left(\alpha_{i,k} \right), \ k = 1, \ldots, K, \\ y &\sim p \left(y | x; \theta, z \right) \propto \exp \left(-\ell \left(f \left(x; \theta, z \right), y \right) \right), \end{aligned} \tag{1}$$

We can estimate the parameters $W := (\mu, \sigma, \alpha)$ via maximum log-likelihood estimation

$$\max_W \ \widehat{\mathbb{E}}_{x,y} \left[\log \int p \left(y | x; \theta, z \right) p \left(z; \alpha \right) p \left(\theta; \mu, \sigma \right) dz d\theta \right]. \tag{2}$$

However, the MLE is intractable due to the integral. We consider the ELBO, *i.e.*,

$$\max_W \max_{q(z), q(\theta)} \ -\widehat{\mathbb{E}}_{x,y} \mathbb{E}_{z \sim q(z), \theta \sim q(\theta)} [\ell \left(f \left(x; \theta, z \right), y \right)] - KL \left(q || p \right), \tag{3}$$

whose $p \left(z \right) = \prod_{k=1}^K \prod_{i=1}^{k-1} \mathcal{C}ategorial \left(z_{i,k} \right) = \prod_{k=1}^K \prod_{i=1}^{k-1} \prod_{l=1}^L \left(\alpha_{i,k}^l \right)^{z_{i,k}^l}$. As shown in Zellner (1988), the optimal solution of (3) in all possible distributions will be the posterior. With such a model, architecture learning can be recast as Bayesian inference.

2.1 Bayesian Meta-network Architecture Learning

Based on the Bayesian view of the architecture search, we can easily extend it to the few-shot meta-learning setting, where we have many tasks, *i.e.*, $\mathcal{D}_t = \{ x_i^t, y_i^t \}_{i=1}^n$. We are required to learn the neural network architectures and the corresponding parameters jointly while taking the task dependences on neural network structure into account.

We generalize the model (1) to the multi-task setting as follows. For the t-th task, we design the model following (1). Meanwhile, the hyperparameters, *i.e.*, (μ, σ, α), are shared across all the tasks. In other words, the layers and architecture priors are shared between tasks. Then, we have the MLE as

$$\max_W \ \widehat{\mathbb{E}}_{\mathcal{D}_t} \widehat{\mathbb{E}}_{(x,y) \sim \mathcal{D}_t} \left[\log \int p \left(y | x; \theta, z \right) p \left(z; \alpha \right) p \left(\theta; \mu, \sigma \right) dz d\theta \right] \tag{4}$$

Similarly, we exploit the convexity of log-sum-exp to achieve the ELBO. However, due to the structures induced by sharing across the tasks, the posteriors for (z, θ) have special dependency, *i.e.*,

$$\max_W \widehat{\mathbb{E}}_{\mathcal{D}_t} \left(\max_{q(z|\mathcal{D}), q(\theta|\mathcal{D})} \ \widehat{\mathbb{E}}_{(x,y) \sim \mathcal{D}_t} \mathbb{E}_{z \sim q(z|\mathcal{D}), \theta \sim q(\theta|\mathcal{D})} \left[-\ell \left(f \left(x; \theta, z \right), y \right) \right] - KL \left(q || p \right) \right). \tag{5}$$

With the variational posterior distributions, $q \left(z | \mathcal{D} \right)$ and $q \left(\theta | \mathcal{D} \right)$, introduced into the model, we can directly generate the architecture and its corresponding weights based on the posterior. In a sense, the posterior can be understood as the neural network predictive model.

3 Variational Inference by Optimization Embedding

The design of the parameterization of the posterior $q \left(z | \mathcal{D} \right)$ and $q \left(\theta | \mathcal{D} \right)$ is extremely important, especially in our case where we need to model the dependence w.r.t. the task distributions $\mathcal{D}$. We bypass this problem by applying parametrized coupled variational Bayes (CVB) (Dai et al., 2018), which generates the parameterization automatically using *optimization embedding* (Dai et al., 2018).

Specifically, we assume the $q \left(\theta | \mathcal{D} \right)$ is Gaussian and the $q \left(z | \mathcal{D} \right)$ is product of the Categorical distribution. To ensure the gradient is still valid, we approximate the categorical z with Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2016). Therefore, we have

$$q_\psi \left(\theta | \mathcal{D} \right) = \mathcal{N} \left(\psi_\mu, \psi_\sigma \right), \quad q_\phi \left(z_{i,k} | \mathcal{D} \right) = \Gamma \left(r \right) \tau^{L-1} \left(\sum_{l=1}^L \frac{\pi_{\mathcal{D}, \phi_{i,k}^l, l}}{\left(z_{i,k}^l \right)^\tau} \right)^{-r} \prod_{i=1}^r \left(\frac{\pi_{\mathcal{D}, \phi_{i,k}^l, l}}{\left(z_{i,k}^l \right)^{\tau+1}} \right), \tag{6}$$

Then, we can sample (θ, z) by following,

$$\theta_{\mathcal{D}}(\epsilon, \psi) = \psi_{\mathcal{D},\mu} + \epsilon\psi_{\mathcal{D},\sigma}, \quad \epsilon \sim \mathcal{N}(0,1), \tag{7}$$

$$z^l_{i,k,\mathcal{D}}(\xi, \phi) = \frac{\exp\left(\left(\phi^l_{\mathcal{D},i,k} + \xi^l\right)/\tau\right)}{\sum_{l=1}^{L}\exp\left(\left(\phi^l_{i,k} + \xi^l\right)/\tau\right)}, \quad \xi^l \sim \mathcal{G}(0,1), \quad l \in \{1,\ldots,L\}, \tag{8}$$

with $\pi_{x,\phi,i} = \frac{\exp(\phi_{x,i})}{\sum_{i=1}^{p}\exp(\phi_{x,i})}$ and $\mathcal{G}(0,1)$ denotes the Gumbel distribution. We emphasize that we do not have any explicit form of the parameters $\phi_{\mathcal{D}}$ and $\psi_{\mathcal{D}}$ yet, which will be derived by optimization embedding. Plugging the formulation into the ELBO (5), we arrive at the objective

$$\widehat{\mathbb{E}}_{\mathcal{D}}\left[\max_{\phi_{\mathcal{D}},\psi_{\mathcal{D}}} \underbrace{\widehat{\mathbb{E}}_{x,y}\mathbb{E}_{\xi,\epsilon}\left[-\ell\left(f\left(x;\theta_{\mathcal{D}}(\epsilon,\psi), z_{\mathcal{D}}(\xi,\phi)\right), y\right)\right] - \log\frac{q_\phi(z|\mathcal{D})}{p(z;\alpha)} - \log\frac{q_\psi(\theta|\mathcal{D})}{p(\theta;\mu,\sigma)}}_{L(\phi_{\mathcal{D}},\psi_{\mathcal{D}};W)}\right]. \tag{9}$$

We follow the parametrized CVB derivation (Dai et al., 2018) for embedding the optimization procedure for (ϕ, ψ). Denoting the $\widehat{g}_{\phi_{\mathcal{D}},\psi_{\mathcal{D}}}(\mathcal{D}, W) = \frac{\partial\widehat{L}}{\partial(\phi_{\mathcal{D}},\psi_{\mathcal{D}})}$ where $\widehat{L}$ is the stochastic approximation for $L(\phi_{\mathcal{D}}, \psi_{\mathcal{D}}; W)$, we have

$$\left[\phi^t_{\mathcal{D}}, \psi^t_{\mathcal{D}}\right] = \eta_t\widehat{g}_{\phi_{\mathcal{D}},\psi_{\mathcal{D}}}(\mathcal{D}, W) + \left[\phi^{t-1}_{\mathcal{D}}, \psi^{t-1}_{\mathcal{D}}\right],$$

We can initialize $\left(\phi^0, \psi^0\right) = W$ which is shared across all the tasks. Alternative choices are also possible, *e.g.*, with one more neural network, $\left(\phi^0, \psi^0\right) = h_V(\mathcal{D})$. After T steps of the iteration, we obtain $\left(\phi^T_{\mathcal{D}}, \psi^T_{\mathcal{D}}\right)$, which leads to $\left(\theta^T_{\mathcal{D}}\left(\xi, \psi^T_{\mathcal{D}}\right), z_{\mathcal{D}}\left(\xi, \phi^T_{\mathcal{D}}\right)\right)$ by (7). In other words, we derive the concrete parameterization of $q(\theta|\mathcal{D})$ and $q(z|\mathcal{D})$ automatically by unfolding the optimization steps. Plugging the ultimate parameterization into $L(\phi_{\mathcal{D}}, \psi_D, W)$, we have

$$\max_{W,V}\ \widehat{\mathbb{E}}_{\mathcal{D}}\widehat{\mathbb{E}}_{x,y}\mathbb{E}_{\xi,\epsilon}\left[-\ell\left(f\left(x;\theta^T_{\mathcal{D}}(\epsilon,\psi), z^T_{\mathcal{D}}(\xi,\phi)\right), y\right) - \log\frac{q_{\phi^T_{\mathcal{D}}}(z|\mathcal{D})}{p(z;\alpha)} - \log\frac{q_{\psi^T_{\mathcal{D}}}(\theta|\mathcal{D})}{p(\theta;\mu,\sigma)}\right]. \tag{10}$$

which can be optimized by stochastic gradient ascent for learning W.

The instantiated algorithm from optimization embedding in this case shares some similarities to second-order MAML (Finn et al., 2017) and DARTS (Liu et al., 2018) algorithms. Both of these two algorithm unroll the stochastic gradient step. However, with the introduction of the Bayesian view, we can exploit the rich literature for the approximation of the distributions on discrete variables. More importantly, we can easily share both the architecture and weights across many tasks. Finally, it establishes the connection between the heuristic MAML algorithm to Bayesian inference, which can be of independent interest. The psuedocode for the concrete algorithm for few-shot Bayesian meta-Architecture SEarch (BASE) can be found in Appendix A.

4 Experiments

All experiments were run on a commonly used benchmark for few-shot learning, the Mini-Imagenet dataset as proposed in Ravi and Larochelle (2017), specifically on the 5-way classification 5-shot learning problem.

In practice, it is very computationally expensive to run architecture search on full sized models which is required to directly adapt weights and architectures with state-of-the-art accuracies. Thus, we use our algorithm to conduct architecture search over the cell architecture search space (Zoph et al., 2017) in which networks structures are composed of repeated cells which share the same architecture, but have different weights. Each node in the cell can be assumed to have a layer connection to every other node after it in the cell. With the inclusion of skip connections and 'no connections' in the choice of operations, we lose no generality in the cell architecture space.

Our algorithm can be used to discover the optimal cells for few-shot learning through our search over smaller networks. We can then evaluate performance on full size networks by transferring the cells to an expanded network. The full sized network is trained on the few-shot learning problem using second-order MAML (Finn et al., 2017) Search and full training were run twice for each method.

A variation of our algorithm was also run using a simple softmax approximation of the Categorical distribution as proposed in Liu et al. (2018) to test the effect of the Gumbel-Softmax architecture parameterization.

Architecture	5-shot Test Accuracy	Parameters
BASE (Softmax)	$65.4 \pm 0.74\%$	1.2M
BASE (Gumbel-Softmax)	$\mathbf{66.2 \pm 0.7\%}$	1.2M
DARTS Architecture	$63.95 \pm 1.1\%$	1.6M
MAML Finn et al. (2017)	$63.11 \pm 0.92\%$	-
REPTILE (Nichol and Schulman, 2018)	$65.99 \pm 0.58\%$	-

Table 1: Comparison of few-shot learning baselines against MAML (Finn et al., 2017) using the architectures found by our BASE (Bayesian meta-Architecture SEarch) algorithm on few-shot learning on the Mini-Imagenet dataset.

Diagrams of the network motifs used for the search network and the full networks can be found in Appendix B.1. More details about the architecture space can be found in Appendix B. An example of one of the best cell architectures found by BASE can be found in Appendix B.2.

4.1 Results

We conduct a comparison between the architectures found by Bayesian meta-Architecture SEarch (BASE), the architecture found by the DARTS (Liu et al., 2018) algorithm which is optimized for CIFAR10 classification, and results from literature using MAML (Finn et al., 2017) and REPTILE (Nichol and Schulman, 2018) fast adaptation methods. The REPTILE (Nichol and Schulman, 2018) algorithm is an useful comparison since it achieved competitive results in few-shot learning using a derivative based weight adaptation method similar to 1st order MAML (Finn et al., 2017). However, it should be noted that they used significantly more (50) inner optimization steps during evaluation than the original MAML paper(10) (Finn et al., 2017) or our experiments(5). While the DARTS architecture(Liu et al., 2018) was optimized for a different problem, it is strongly comparible as a generalized network since our cell search spaces are identical.

As shown in Table 1, our searched architectures achieved significantly better average testing accuracies than our baselines on five-shot learning on the Mini-Imagenet dataset in the same architecture space. The DARTS architecture also achieved results which were significantly better than that found in the original MAML baseline Finn et al. (2017) showing some transferability between CIFAR10 and meta-learning on Mini-Imagenet. The DARTS architecture, however also had significantly more parameters than our found architectures and trained significantly slower. The Gumbel-Softmax Meta-network parameterization also found better architectures than the simple softmax parameterization, but it should be noted that the variance of results of the MAML algorithm are rather high. Using our algorithm to simultaneously learn the architecture and weight distribution for multiple tasks, we were able to find architectures which achieved state of the art performance for quick adaptation methods on this meta-learning task. Compared to other meta-learning algorithms, the meta-network shares more structures and exploits more information across all the tasks.

5 Conclusion

In this paper we considered the problem of learning a meta-network which can simultaneously adapt both its neural network architecture and weights to many tasks. We proposed a Bayesian view of architecture search, and generalized the model to the meta-learning setting. With the optimization embedding technique (Dai et al., 2018), we automatically incorporated the related information into the parameterization of the posterior. This allowed us to find architectures which achieved state-of-the-art performances on the few-shot Meta-learning problem.

In the future, using recent developments in Meta-learning such as using the Reptile approximation for multistep MAML, it may be possible learn a full-sized meta-network and directly adapt the network to achieve state-of-the-art performance on new tasks without retraining.

References

Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In *AAAI*, 2018a.

Han Cai, Jiacheng Yang, Weinan Zhang, Song Han, and Yong Yu. Path-level network transformation for efficient architecture search. *arXiv preprint arXiv:1806.02639*, 2018b.

Bo Dai, Hanjun Dai, Niao He, Weiyang Liu, Zhen Liu, Jianshu Chen, Lin Xiao Xiao, and Le Song. Coupled variational bayes via optimization embedding. In *NIPS*, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016. URL http://arxiv.org/abs/1603.05027.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations*, 2017.

Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. *arXiv preprint arXiv:1802.01548*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Arnold Zellner. Optimal Information Processing and Bayes's Theorem. *The American Statistician*, 42(4), November 1988.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1707.07012*, 2(6), 2017.

Appendix

A Psuedocode for Bayesian Meta-architecture Search for Few-shot Learning

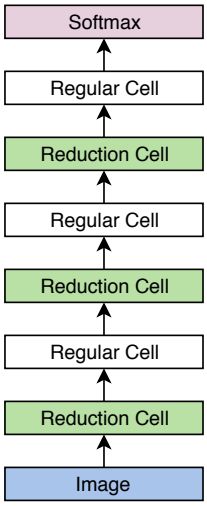
procedure BAYESIAN META-ARCHITECTURE SEARCH FOR FEW-SHOT LEARNING
  Initialize meta-network parameters W
  for iteration in 1 to R **do**
    Sample T tasks $\mathcal{D}$
    for $\mathcal{D}_t$ in $\mathcal{D}$ **do**
      Sample K examples (x_t, y_t) for $\mathcal{D}_t$
      Let $W'_{i,0} = W$
      for $u \leftarrow 0$ to N-1 **do**
        Sample iid g from Gumbel(0,1)
        Adapt parameters with SGD $W'_{i,u+1} = W'_{i,u} - \nabla_{W'_{i,u}} L(f(x; W'_{i,u}, g), y)$
    Sample K examples (x'_t, y'_t) for each $\mathcal{D}_t$
    Sample iid g from Gumbel(0,1)
    update $W \leftarrow W - \nabla_W \sum_{\mathcal{D}_i} \ell(f(x'_i; W'_{i,N}, g), y'_i)$

B Architecture Space Details

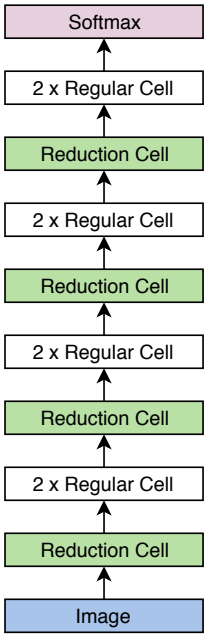
For comparability in architectures, the particular search space used is very similar to that used in Liu et al. (2018) and includes the same operation space: 3×3, 5×5, 7×7 depth-wise separable convolutions, 3×3 and 5×5 dilated depth-wise separable convolutions, 3×3 max pooling, 3×3 average pooling, a 1×7 followed by a 7×1 convolution, skip connections, and no connection. In our search, each cell is made up of a total of six nodes with 2 input nodes. The input to each normal cell is the output from the previous 2 cells and each reducing cell uses two copies of the input from the previous cell. The output for each cell is the concatenated output from all 4 non-input nodes in the cell. Following the same methods as Liu et al. (2018); Zoph et al. (2017), non-dilated depth-wise separable convolutions were applied twice, all depth-wise separable convolutions did not have batch-norms between the grouped and 1x1 convolutions, convolutions had RELUs and batch-norms applied in ReLU-Conv-BN order, and all operations were padded as necessary to preserve spatial resolution as to only be reduced by the reducing layers whose first operations were applied with a stride of 2.

To make the cell space comparable with previous work Liu et al. (2018); Zoph et al. (2017), We use this method to derive the discrete cells for for transferring to the larger network. For each node, we choose the 2 operations to it with the highest probability of being sampled which are coming from different source nodes.

B.1 Motifs for Scalable Architectures

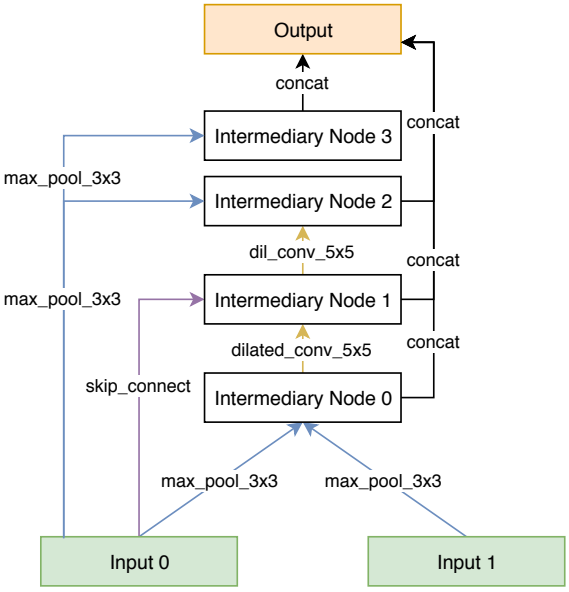


Motif for Search Network

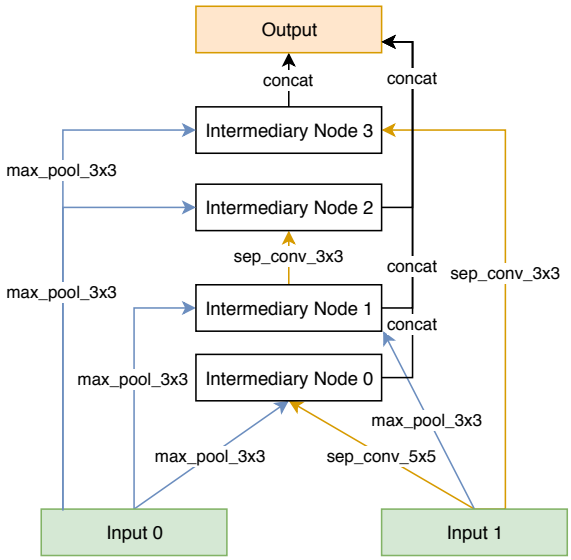


Motif for Full Network.

B.2 Sample Top Found Cell Architectures from BASE search



Cell Architecture for sample normal cell



Cell Architecture for sample reduction cell

B.3 Training Details

In our experiments on the Mini-Imagenet dataset, only the 64 training classes were used during training. The 12 validation classes were ignored, and evaluation was conducted on the 24 testing classes.

Search was run for 10000 iterations. For each iteration, the meta-network was updated with the combined gradients from $T = 2$ randomly sampled tasks. For each task $N = 4$ steps of inner optimization were run. For

the full training, all network architectures were trained with the same setting on the 5-shot learning problem using the second-order MAML algorithm Finn et al. (2017). The full training was run for 30000 iterations. Similarly, for each iteration, the network was again updated with the combined gradients from 2 randomly sampled tasks, but each task was optimized with 5 steps of inner optimization for second-order MAML.