MetaPruning: Meta Learning for Automatic Neural Network Channel Pruning

汇报人：严阳春

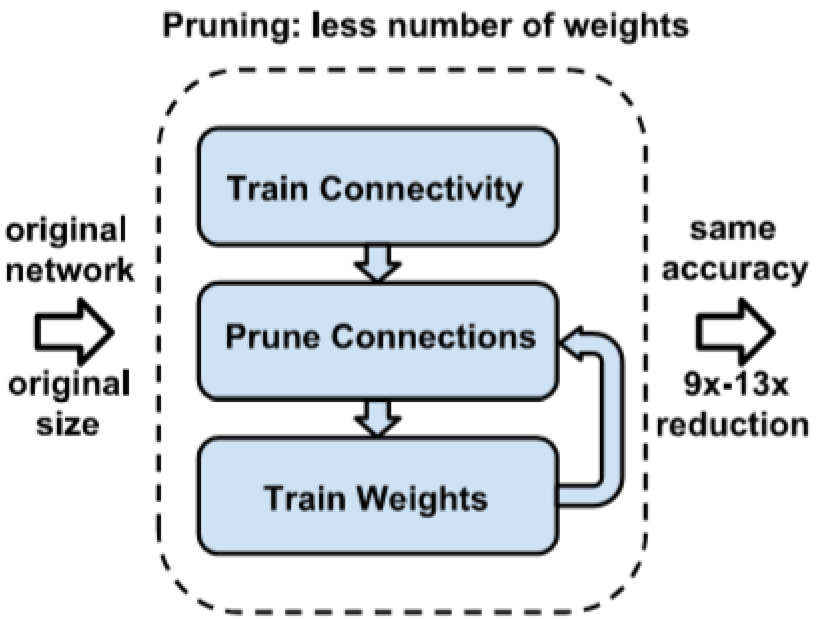
背景：

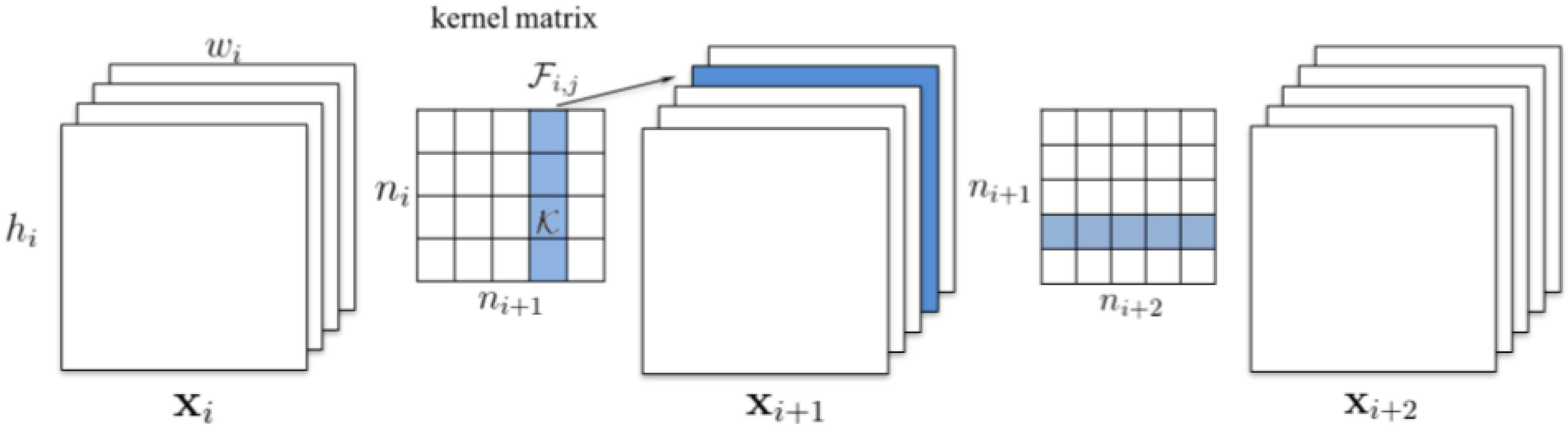
目前的深度神经网络模型对运行平台的存储和计算能力要求较大 ，为了使人工智能应用更加广泛，通过压缩和加速原始深度网络模型，使之直接应用于移动嵌入式设备端，将成为一种有效的解决方案 。目前主流的方法如下：

Category	Explanation	Applications (CONV，FC)	Training from scratch or pre-trained model	Combination
Parameter Pruning	Pruning unsalient parameters by measuring the importance of parameters	Both	Pre-trained model	Support，often with parameter quantization and low-rank decomposition
Parameter Sharing	Reducing redundant parameters using hashing or quantization methods	Both	Both	Support，often with parameter pruning
Low-rank Decomposition	Decomposing generalized convolution into a sequence of matrix/tensor-based convolutions with fewer parameters	Both	Both	Support，often with parameter pruning
Designing Compact Convolutional Filters	Reducing the storage and computation cost by designing compact convolutional filters	CONV	Training from scratch	
Knowledge Distillation	Training a compact neural network with distilled knowledge of a large model	Both	Training from scratch	

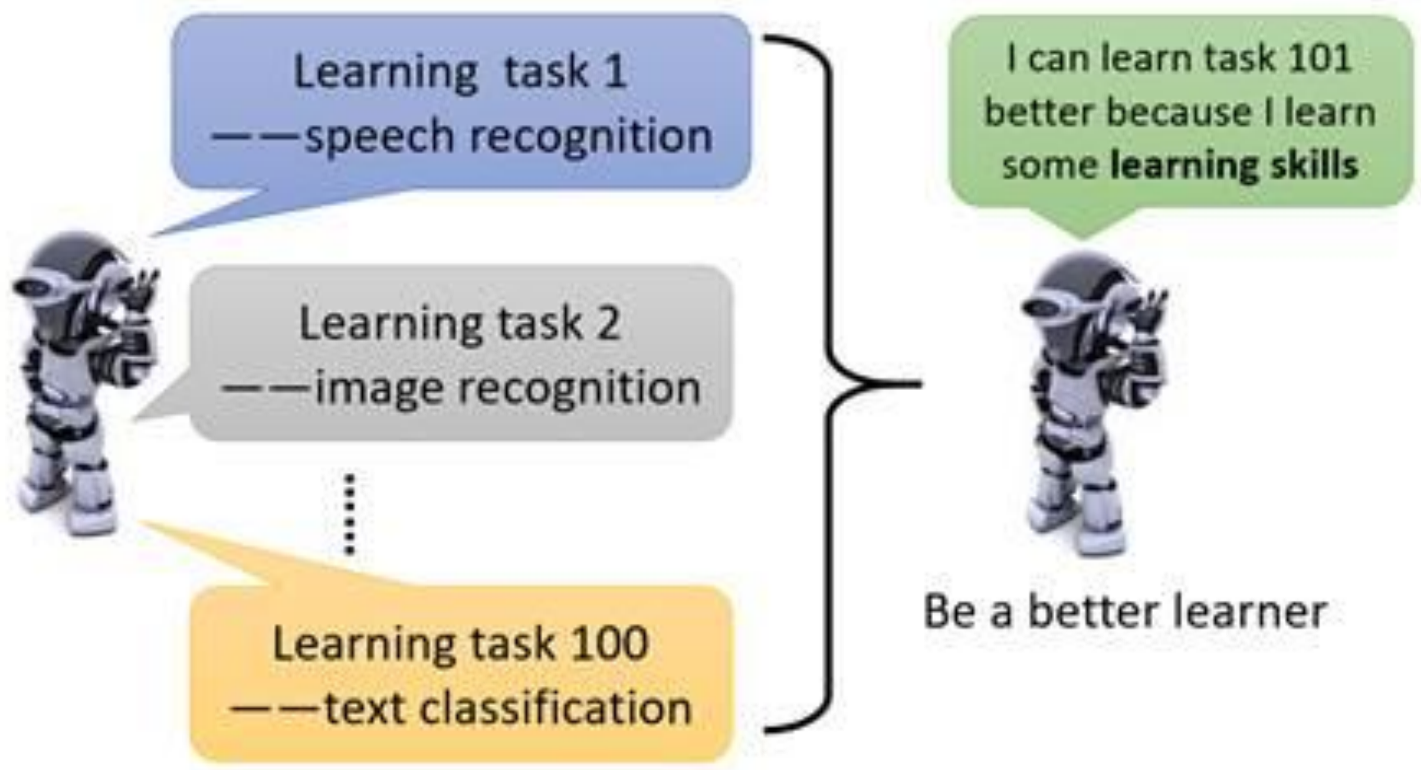
模型剪枝技术

网络/参数剪枝是通过对已有的训练好的深度网络模型移除冗余的、信息量少的权值，从而减少网络模型的参数，进而加速模型的计算和压缩模型的存储空间。不仅如此，通过剪枝网络，能防止模型过拟合。剪枝有不同的剪枝粒度方法和不同的判断权值重要性的方法。



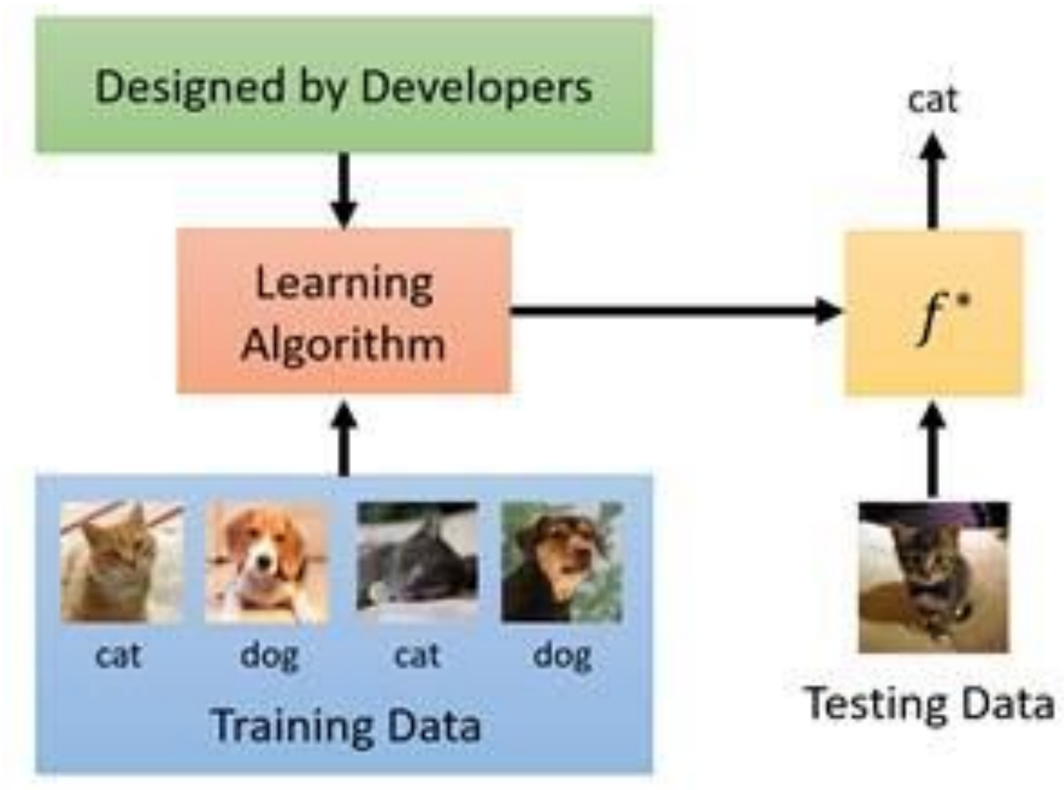


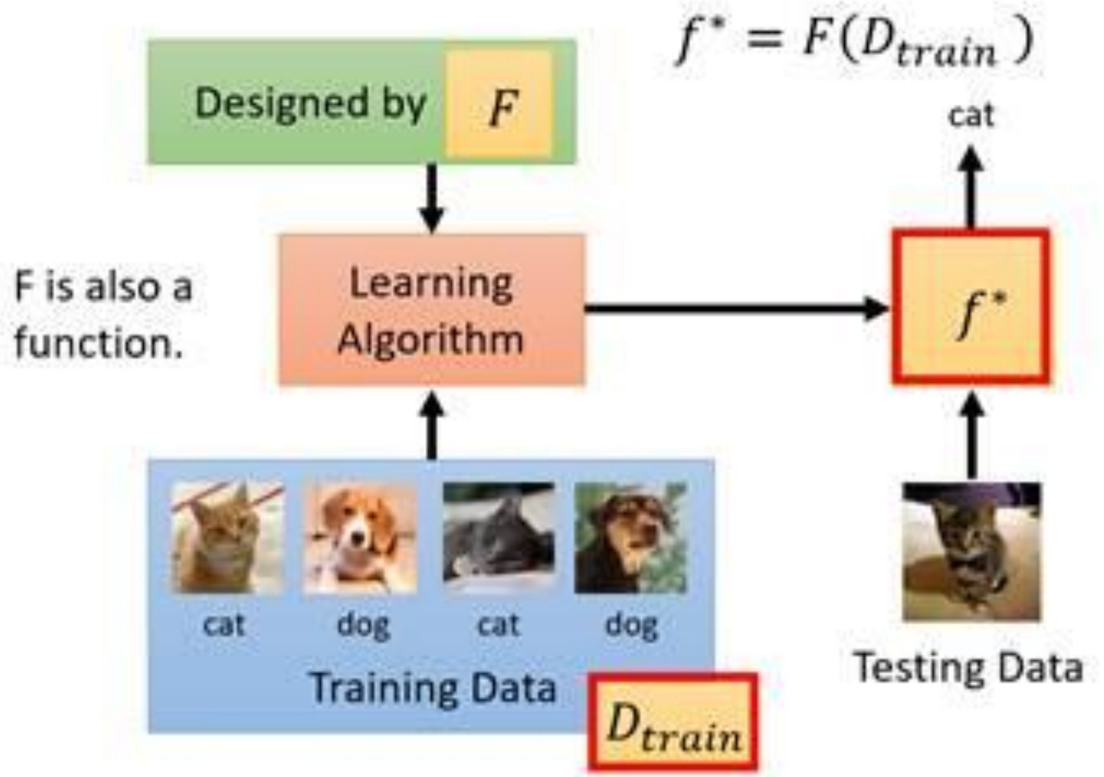
随着相关研究的深入，特别是在对于剪枝后，权重和结构对于模型的重要性比较，产生了一系列的具有创新性的猜想假设和实验方法。例如最近有一篇论文《Rethinking the Value of Network Pruning》指出：**剪枝后的权重并不重要，结构相对重要**。遵循这个思想，论文作者提出了基于meta learning 的自动剪枝技术。



Meta Learning (元学习)

Meta Learning = learning to learn





Machine Learning：根据资料找一个函数f的能力

Meta Learning：根据资料找一个找一个函数f的函数F的能力

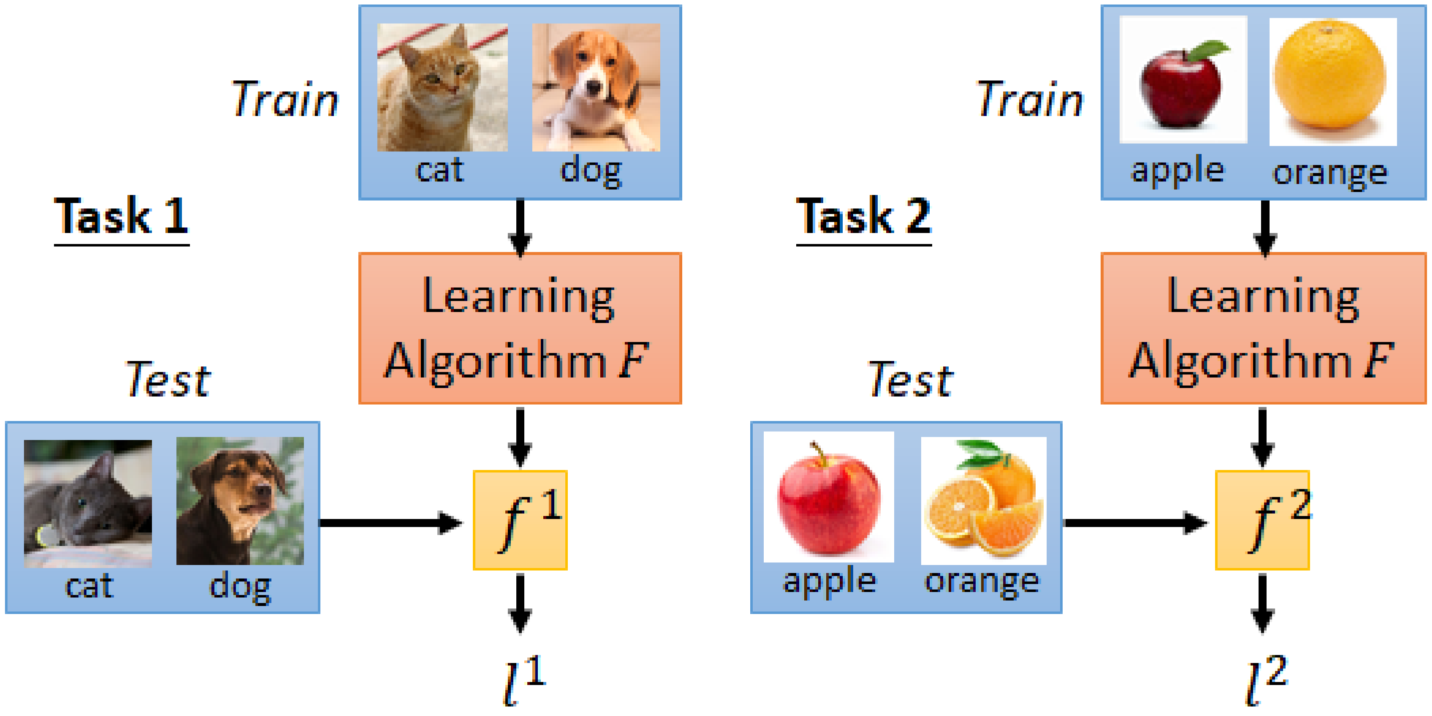
Meta Learning

$$L(F) = \sum_{n=1}^{N} l^n$$

N tasks

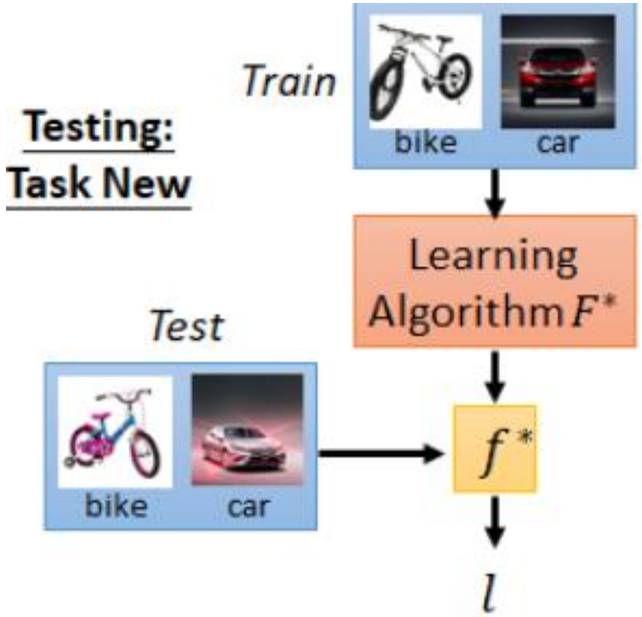
Testing loss for task n after training

- Defining the goodness of a function F



- Find the best function F^*

$$F^* = arg\,\min_F L(F)$$



MetaPruning

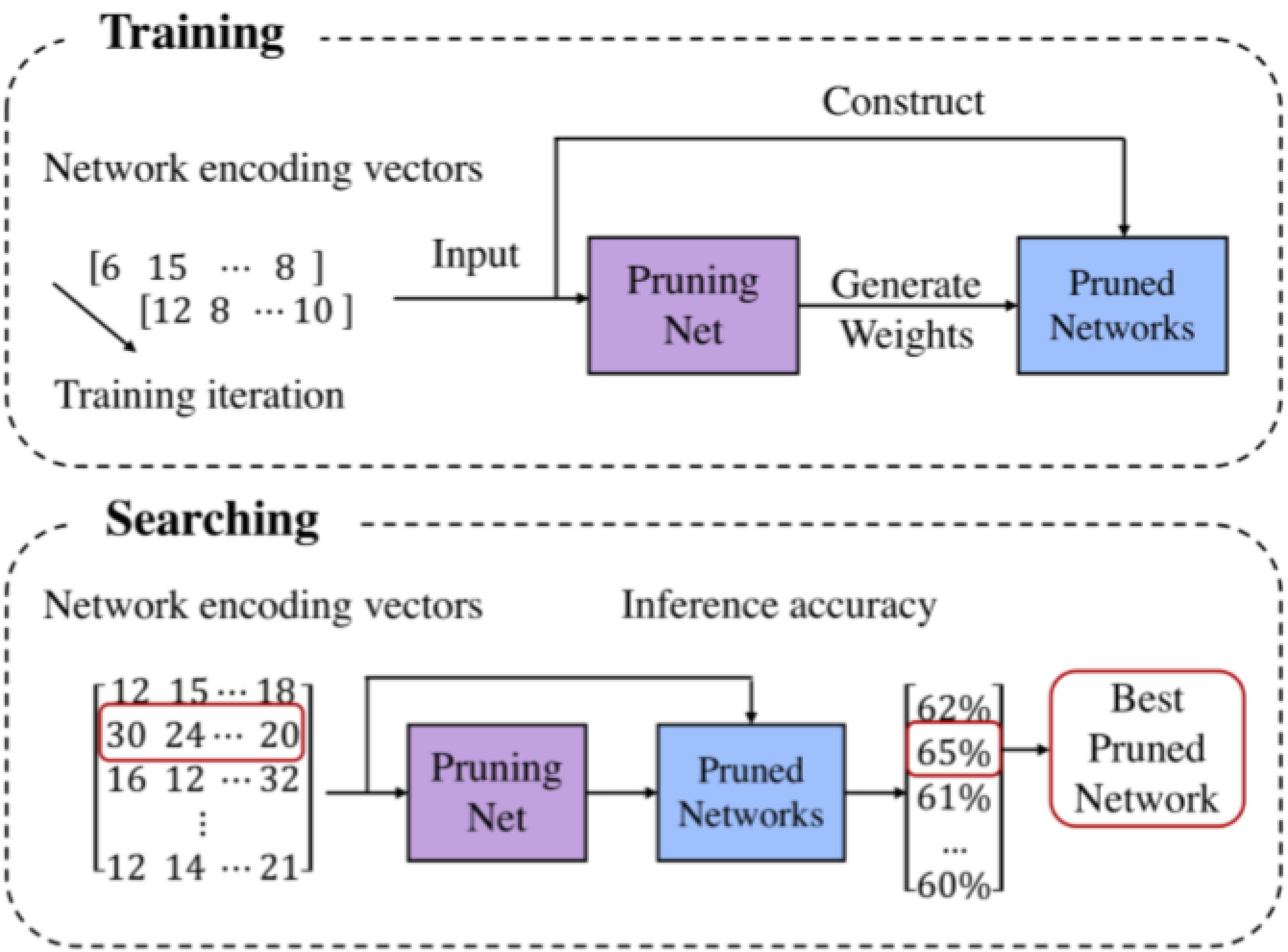
传统模型剪枝：

•训练一个参数量较多的大网络

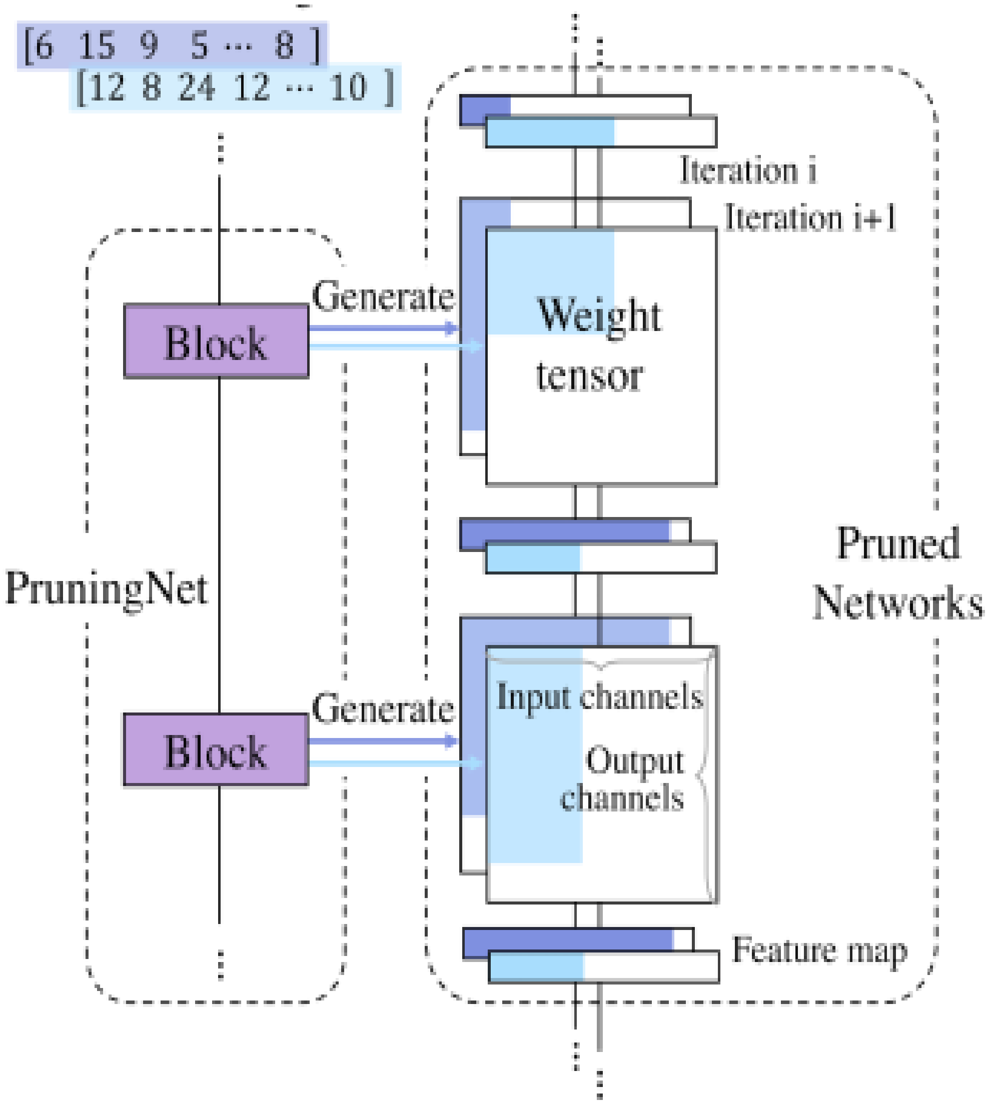
•将不重要的权重参数剪掉

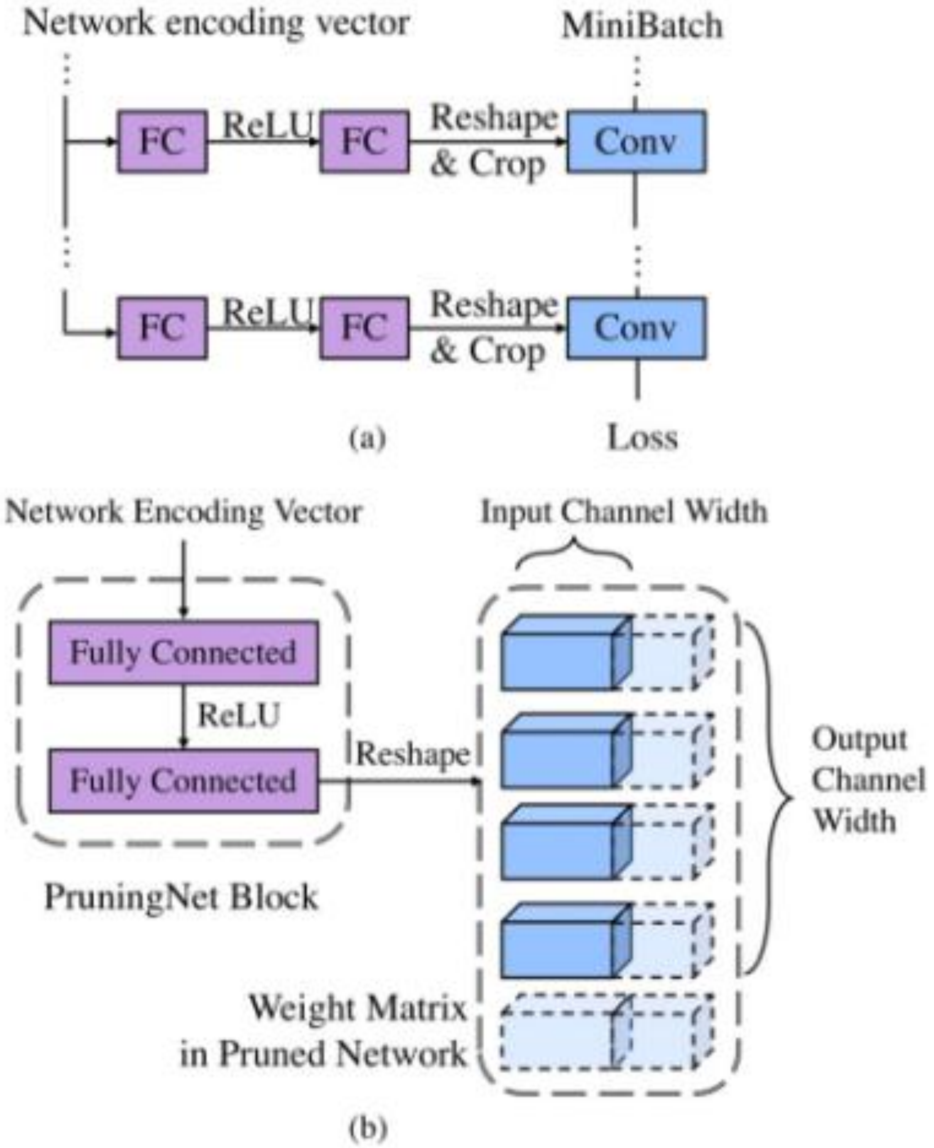
•剪枝后的小网络做 fine tune

MetaPruning：



Training


[6 15 9 5 ··· 8]
[12 8 24 12 ··· 10]
Iteration i
Iteration i+1
Generate
Block
Weight
tensor
PruningNet
Pruned
Networks
Generate
Block
Input channels
Output
channels
Feature map


Network encoding vector
MiniBatch
FC
ReLU
FC
Reshape
& Crop
Conv
FC
ReLU
FC
Reshape
& Crop
Conv
Loss
(a)
Network Encoding Vector
Input Channel Width
Fully Connected
ReLU
Fully Connected
Reshape
Output
Channel
Width
PruningNet Block
Weight Matrix
in Pruned Network
(b)

Searching

在训练好PruningNet后，获得了能产生相应剪枝结构的权重的能力，采用进化算法去搜索最好的剪枝后的网络结构。然后再从头训练这个结构组成的网络。

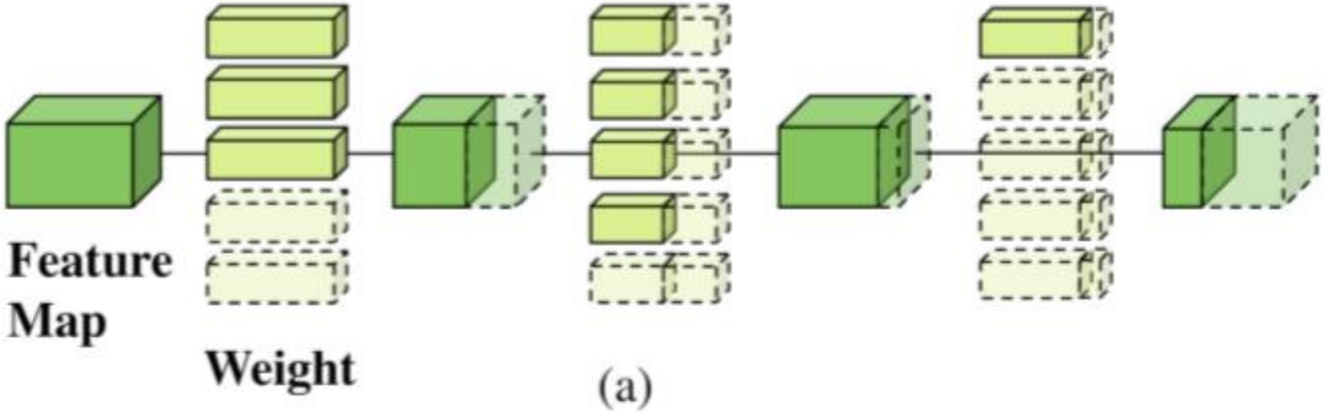
Algorithm 1 Evolutionary Search Algorithm

Hyper Parameters: Population Size: $\mathcal{P}$, Number of Mutation: $\mathcal{M}$, Number of Crossover: $\mathcal{S}$, Max Number of Iterations: $\mathcal{N}$.
Input: PruningNet: $PruningNet$, Constraints: $\mathcal{C}$.
Output: Most accurate gene: $\mathcal{G}_{top}$.

```
G_0 = Random(P), s.t. C;
G_topK = ∅;
for i = 0 : N do
    {G_i, accuracy} = Inference(PruningNet(G_i));
    G_topK, accuracy_topK = TopK({G_i, accuracy});
    G_mutation = Mutation(G_topK, M), s.t. C;
    G_crossover = Crossover(G_topK, S), s.t. C;
    G_i = G_mutation + G_crossover;
end for
G_top1, accuracy_top1 = Top1({G_N, accuracy});
return G_top1;
```

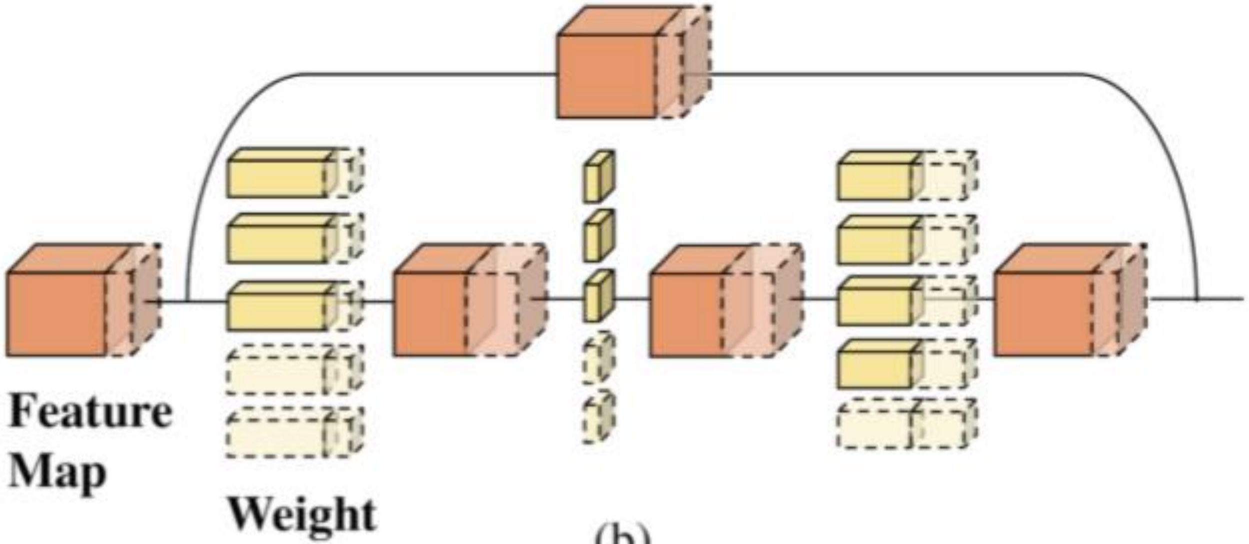
Experiment

(a)是针对没有shortcut的网络结构，例如MobileNet V1



(a)

(b)是针对有shortcut的网络结构，例如MobileNet V2 , ResNet



(b)

Pruning under FLOPs constraint

Uniform Baselines			MetaPruning	
Ratio	Top1-Acc	FLOPs	Top1-Acc	FLOPs
1×	70.6%	569M	–	–
0.75×	68.4%	325M	**70.9%**	324M
0.5×	63.7%	149M	**66.1%**	149M
0.25×	50.6%	41M	**57.2%**	41M

此方法在MobileNet V1上的效果

Uniform Baselines		MetaPruning	
Top1-Acc	FLOPs	Top1-Acc	FLOPs
74.7%	585M	–	–
72.0%	313M	**72.7%**	291M
67.2%	140M	**68.2%**	140M
66.5%	127M	**67.3%**	124M
64.0%	106M	**65.0%**	105M
62.1%	87M	**63.8%**	84M
54.6%	43M	**58.3%**	43M

此方法在MobileNet V2上的效果

Network		FLOPs	Top1-Acc
Uniform Baseline	1.0× ResNet-50	4.1G	76.6%
	0.75× ResNet-50	2.3G	74.8%
	0.5 × ResNet-50	1.1G	72.0%
Traditional Pruning	SFP[20]	2.9G	75.1%
	ThiNet-70 [38]	2.9G	75.8%
	ThiNet-50 [38]	2.1G	74.7%
	ThiNet-30 [38]	1.2G	72.1%
	CP [22]	2.0G	73.3%
MetaPruning - 0.85×ResNet-50		3.0G	**76.2%**
MetaPruning - 0.75×ResNet-50		2.0G	**75.4%**
MetaPruning - 0.5 ×ResNet-50		1.0G	**73.4%**

此方法和其他方法在ResNet上的比较

Network	FLOPs	Top1-Acc
0.75x MobileNet V1 [24]	325M	68.4%
NetAdapt [52]	284M	69.1%
AMC [21]	285M	70.5%
MetaPruning	281M	**70.6%**
0.75x MobileNet V2 [46]	220M	69.8%
AMC [21]	220M	70.8%
MetaPruning	217M	**71.2%**

此方法和其他基于AutoML的方法比较

Pruning under latency constraint

在一块Titan Xp，batch size为32的情况下运行相应网络结构的时间
如下表：

Uniform Baselines			MetaPruning	
Ratio	Top1-Acc	Latency	Top1-Acc	Latency
1×	70.6%	0.62ms	–	–
0.75×	68.4%	0.48ms	**71.0%**	0.48ms
0.5×	63.7%	0.31ms	**67.4%**	0.30ms
0.25×	50.6%	0.17ms	**59.6%**	0.17ms

MobileNet V1

Uniform Baselines			MetaPruning	
Ratio	Top1-Acc	Latency	Top1-Acc	Latency
1.4×	74.7%	0.95ms	–	–
1×	72.0%	0.70ms	**73.2%**	0.67ms
0.65×	67.2%	0.49ms	**71.7%**	0.47ms
0.35×	54.6%	0.28ms	**64.5%**	0.29ms

MobileNet V2

总结：

这篇文章从“剪枝后的权重不重要”的前提出发，将剪枝和MetaLearning结合，提出了PruningNet为剪枝后的网络预测权重。并在编码网络信息的encode vector 的状态空间进行搜索，找到给定约束条件下的最优网络结构，在ImageNet数据集和ResNet/MobileNet-v1/v2上取得了比之前剪枝算法更好的效果。