Channel Planting for Deep Neural Networks using Knowledge Distillation

Kakeru Mitsuno, Yuichiro Nomura and Takio Kurita

Hiroshima University, Japan

HIROSHIMA UNIVERSITY

Kurita Laboratory

Abstract

In recent years, deeper and wider neural networks have shown excellent performance in computer vision tasks, while their enormous amount of parameters results in increased computational cost and overfitting. Several methods have been proposed to compress the size of the networks without reducing network performance, such as network pruning and knowledge distillation. The performance of the smaller network obtained by these methods is bounded by the predefined network. In this paper, we present a novel incremental training algorithm for deep neural networks called ***planting***. Our planting can search the optimal network architecture with smaller number of parameters for improving the network performance by augmenting channels incrementally to layers of the initial networks while keeping the earlier trained parameters fixed. Also, we propose using the knowledge distillation method for training the channels planted. By transferring the knowledge of deeper and wider networks, we can grow the networks effectively and efficiently.

We evaluate the effectiveness of the proposed method on different datasets such as CIFAR-10/100 and STL-10. For the STL-10 dataset, we show that we are able to achieve comparable performance with only 7% parameters compared to the larger network and reduce the overfitting caused by a small amount of the data.

Proposed Method

Our planting approach consists of the following training Processes (show in Fig.1)

(0) Training a large network as the teacher network.

(1) Training a small network with fewer channels of each layer by a standard classification training method.

(2) Adding channels to a layer on the small network by using a knowledge distillation method with the teacher network while keeping the earlier trained parameters fixed.

(3) Repeat (2) the number of layers times

(4) Selecting a planted network with the smallest validation loss as the next base network for planting

(5) Repeating (4) while reducing the classification loss than the previous network

Experiments

We have performed experiments using CIFAR-10/100 and STL-10. The structure of networks are shown in Table.1.

Table.1 The Structure of Networks

For CIFAR-10/100	For STL-10
ReLU(conv1(kernel=3))	ReLU(conv1(kernel=3))
max pooling(2*2)	max pooling(2*2)
ReLU(conv2(kernel=3))	ReLU(conv2(kernel=3))
max pooling(2*2)	max pooling(2*2)
ReLU(conv3(kernel=3))	ReLU(conv3(kernel=3))
ReLU(conv4(kernel=3))	max pooling(2*2)
ReLU(conv5(kernel=3))	ReLU(conv4(kernel=3))
max pooling(2*2)	ReLU(conv5(kernel=3))
ReLU(fc1())	max pooling(2*2)
output=fc2()	ReLU(fc1())
	output=fc2()

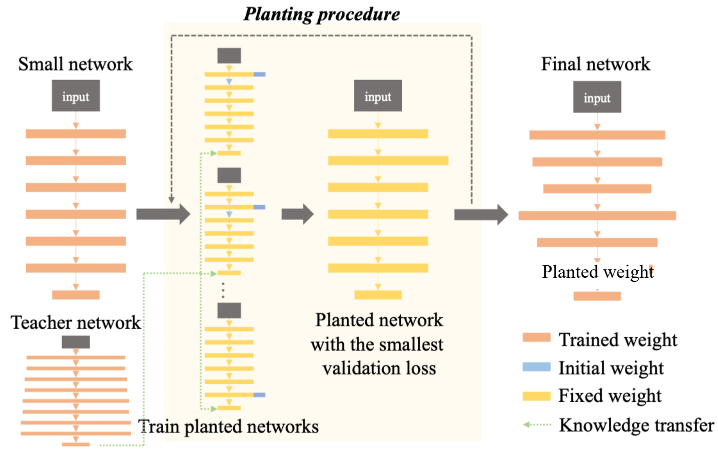

Fig.1 Illustration of Planting Procedure on a typical DNNs

The results on each datasets are shown in Table.2,3 and 4.

Table.2 The Results on CIFAR-10

Network	Params	Test Err.	Test Acc.	Loss func
Teacher[128]	857.5K	0.5007	88.10%	CELoss
Student[128]		0.3823	88.51%	KLLoss
Initial Network (Student[8])	20.4K	0.8300	71.55%	CELoss
		0.8245	71.69%	KLLoss
Student[16]	43.9K	0.6071	79.42%	CELoss
		0.6108	79.23%	KLLoss
Student[32]	104.8K	0.4898	84.03%	CELoss
		0.4791	84.02%	KLLoss
Student[64]	282.0K	0.4431	86.83%	CELoss
		0.4103	86.80%	KLLoss
Ours	40.6K	0.4825	84.35%	KLLoss

Table.3 The Results on CIFAR-100

Network	Params	Test Err.	Test Acc.	Loss func
Teacher[128]	869.1K	2.5010	57.76%	CELoss
Student[128]		1.6232	60.05%	KLLoss
Student[8]	32.0K	2.5280	36.53%	CELoss
		2.5053	36.90%	KLLoss
Initial Network (Student[16])	55.5K	2.1190	45.45%	CELoss
		2.0679	46.66%	KLLoss
Student[32]	116.5K	1.9022	52.15%	CELoss
		1.7805	53.72%	KLLoss
Student[64]	293.6K	1.9510	55.74%	CELoss
		1.6707	57.71%	KLLoss
Ours	78.5K	1.7584	54.31%	KLLoss

Table.4 The Results on STL-10

Network	Params	Test Err.	Test Acc.	Loss func
Teacher[64]	445.8K	1.5360	66.33%	CELoss
Student[64]		1.1807	66.47%	KLLoss
Initial Network (Student[8])	40.8K	1.2776	55.55%	CELoss
		1.2682	54.99%	KLLoss
Student[16]	84.9K	1.2924	59.34%	CELoss
		1.1998	61.10%	KLLoss
Student[32]	186.8K	1.2213	64.57%	CELoss
		1.1712	64.07%	KLLoss
Student[128]	1.2M	1.7612	67.04%	CELoss
		1.1643	67.71%	KLLoss
Ours	82.6K	1.0772	67.12%	KLLoss

Conclusion

We proposed a novel incremental training method called ***planting*** using knowledge transfer, that can train smaller network with excellent performance and find the optimal network architecture automatically. We confirmed that the proposed approach was able to achieve comparable performance with smaller parameters compare to the larger network and reduce the over-fitting.