Shallow Networks for Semantic Segmentation

Ulaş Berk Karlı
Koç University
Istanbul
ukarli16@ku.edu.tr

***Abstract*—Convolutional neural networks that are used in semantic segmentation task are deep neural networks which have a lot of parameters to store and this takes huge memory space and time in testing. With emerging embedded applications of computer vision with deep learning and semantic segmentation creates the need of shallower networks that have similar or same accuracy. In this project the main aim was creating a network that is shallower than it is compared deep network and as accurate as it is.**

Keywords—fcn, semantic, segmentation, Resnet, knowledge distillation

I. Introduction

Convolutional neural networks are used for different task in computer vision. One of these tasks is semantic segmentation. Semantic segmentation is labeling each pixel of an image with a class label. Semantic segmentation task differs from instance segmentation that in semantic segmentation there is no difference in class instances. For example, if there are two cars in an image in semantic segmentation both cars are label with the same car label but in instance segmentation cars are labeled as two different instances of a car class. For this semantic segmentation task most common and fundamental network is Fully Convolutional Networks [4]. These networks are composed of all convolutional layers as the name suggests, there is no fully connected layer after the last convolutional layer up sampling is done to go back to the image size. These fully convolutional networks are deep and for an embedded application or when there are not enough memory space shallower networks are preferred but they are not accurate as deeper networks. There is a solution proposed for this an it is knowledge distillation [2]. This project is done in order to train shallower networks that are as accurate as deep networks by using knowledge distillation for fully convolutional networks.

II. Methods

For knowledge distillation the loss function proposed by Hinton et al. is used. This loss function has two parts one part is using the ground truth labels as usual and the other part is the actual knowledge distillation part. For that we utilize the pre-SoftMax logits z. As proposed, there is a temperature coefficient which softens the probability distributions. When temperature is set to one regular SoftMax values are calculated.

$$q_i = \frac{\exp\,(z_i/T)}{\sum \exp\,(z_j/T)}$$

The resulting class probabilities are used to calculate the knowledge distillation loss which is the difference between the probability distributions of teacher model and the student model. This distance is measured as Kullback-Leibler divergence. Thus, the resulting compound loss used throughout this project is as fallows.

$$L = \alpha T^2 D_{KL}(S \parallel T) + (1-\alpha)(E_S[-\log y])$$

Alpha is the a hyperparameter used for weighted sum of the regular cross entropy loss and the knowledge distillation loss.

There are other approaches to knowledge distillation in semantic segmentation task. In Structured Knowledge Distillation for Dense Prediction [9] paper two more loss functions are proposed to distill some sort of structured knowledge from the networks since in semantic segmentation there are some structured knowledge their base loss is pixel-wise distillation loss which is similar to the loss above but not the same.

III. Experiments

Two major experiments are performed in this project. They are separate since one uses convolutional networks for object classification task and the other experiment uses fully convolutional networks for semantic segmentation. Second experiment is the main experiment of this project.

A. Knowledge Distillation Experiment

First to understand knowledge distillation concept a distillation experiment on a Resnet18 [5] and a 5-layer CNN is performed. A Resnet18 model which is trained on CIFAR10 [1] is used as teacher and a 5-layer CNN is used as student to distill the knowledge of Resnet18 in to the 5-layer CNN. This experiment focused on reproducing the results of Haitong Li [3] project in Stanford course.

Teacher	Resnet18
Student	5-layer CNN
Optimizer	Adam
Batch Size	128
Learning Rate	1e-3
Temperature	20
Alpha	0.9
Number of Epochs	30

B. *Shallow FCN experiment*

For the main experiment we have implemented a student network based on the FastFCN [10] code base is used. The student has a Resnet50 backbone and for the teacher an FCN with Resnet101 backbone is used. Both teacher and student Resnet's are pretrained on ImageNet [7]. The teacher is loaded from torchvision models which is a fully convolutional network trained on COCO dataset [8]. We then implemented the optimizer and the training and validation functions. Then training is done on Pascal VOC dataset [6]. The experiment hyperparameters as follows.

Teacher	FCN with Resnet101
Student	FCN with Resnet50
Optimizer	SGD with Momentum
Batch Size	8
Learning Rate	2e-3 FCN head 2e-4 Resnet backbone
Momentum	0.9
Weight Decay	0.0005
Temperature	20
Alpha	0.9
Number of Epochs	20

Two more experiments under this second experiment are done.

I. Without Weight Decay

II. For 40 epochs

Results of all these experiments are given below.

IV. Results

For the first experiment the metric used for measuring success is accuracy of predictions. For second experiment there are three main metrics but the most common and a good metric for understanding the performance is intersection over union (IoU) or the Jaccard index. This metric is calculated for each class and then mean over these all classes are taken. IoU is calculated as below.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Results of the first experiment is given below.

Network	Accuracy
Resnet18	%95
5-layer CNN	%80

Results of the second experiment is given below.

Network	mIoU
FCN with Resnet101	%63.7
FCN with Resnet50	%57.9
FCN with Resnet50 w/o momentum	%54.2
FCN with Resnet 50 for 40 epochs	%61.4

For visualization of the second experiment following images are created.

In the first figure we can observe an image from the training set and the ground truth for that image. Second figure is the outputs of the fully convolutional networks. Left image in figure two show the output of the teacher network and the image on the right shows the output of the student network before training.

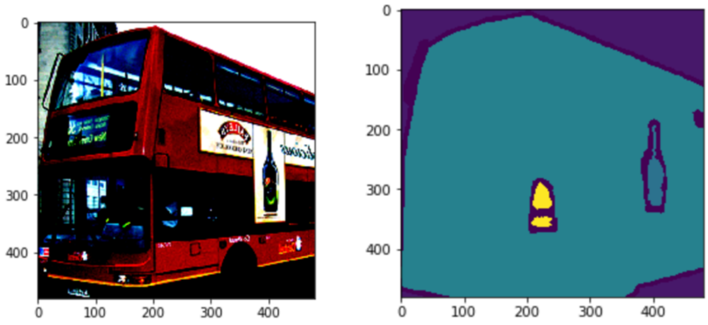

Figure 1: Image from training set and ground truth

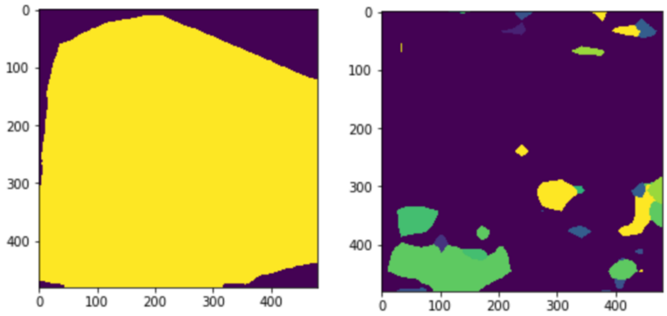

Figure 2: FCN outputs before training

Figure 3: Image from the validation set

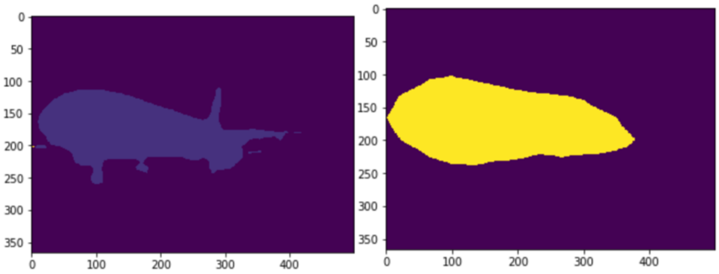

Figure 4: FCN outputs after training

Figure 3 shows a sample from the validation set of the Pascal VOC dataset. Figure 4 shows the FCN outputs after training. In figure 4 image on the left is the output of the teacher network and on the right the output of the student network. These are obtained after training.

V. Conculusion

Results of these experiments show that knowledge distillation is a valid option for semantic segmentation task networks especially fully convolutional networks. First experiment was performed in order to gain knowledge on how to perform knowledge distillation and performing that experiment helped to perform the second experiment. From visualization we can observe that before training student network gave total random outputs but after training, we can see it gets similar to the output of the teacher network. We can also observe that weight decay has no significant effect over results but training for more epochs definitely increases the accuracy of the student network. Hyperparameter optimization could further improve performance so that student mimics the teacher perfectly. Also, bigger batches cloud improve the result but Google Colab do not have enough GPU memory so we could not experiment with it. Overall, we can observe that knowledge distillation even simple pixel wise distillation can be used to train a shallow fully convolutional network.

Code for this project can be found in the following GitHub repo.

- https://github.com/ulaskarli/ShallowNetworks

Acknowledgment

Following GitHub repos are used as code bases for some parts of the project.

- https://github.com/affromero/FCN
- https://github.com/wuhuikai/FastFCN
- https://github.com/irfanICMLL/structure_knowledge_distillation
- https://github.com/peterliht/knowledge-distillation-pytorch

References

[1] A. Torralba and R. Fergus and W. T. Freeman, 80 Million Tiny Images: a Large Database for Non- Parametric Object and Scene Recognition, IEEE PAMI, 2008

[2] Geoffrey Hinton, Oriol Vinyals: "Distilling the Knowledge in a Neural Network", 2015

[3] Haitong Li: "Exploring Knowledge Distillation of Deep Neural Networks for Efficient Hardware Solutions",2018

[4] Jonathan Long, Evan Shelhamer: "Fully Convolutional Networks for Semantic Segmentation", 2014; [http://arxiv.org/abs/1411.4038 arXiv:1411.4038].

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren: "Deep Residual Learning for Image Recognition", 2015; [http://arxiv.org/abs/1512.03385 arXiv:1512.03385].

[6] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg: "ImageNet Large Scale Visual Recognition Challenge", 2014; [http://arxiv.org/abs/1409.0575 arXiv:1409.0575].

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick: "Microsoft COCO: Common Objects in Context", 2014; [http://arxiv.org/abs/1405.0312 arXiv:1405.0312].

[9] Yifan Liu, Changyong Shun, Jingdong Wang: "Structured Knowledge Distillation for Dense Prediction", 2019

[10] https://github.com/wuhuikai/FastFCN