Project2: Deep Learning and Feature Visualization

Zhou Yiyuan

June 2019

1 Introduction

In this task, our group are required to implement the three tasks of multi-category classification, image reconstruction, image segmentation on multi datasets including MNIST, CIFAR-10, CUB200-2011, Pascal VOC 2012 and so on. What's more, contrast experiments and feature visualization are required as well to analyze the influence of neural network's structure.

In our group, Zhou Yiyuan is responsible of image segmentation, Xie Fan and Gu Yuyi are responsible of multi-category, and He Yunfan is responsible of image reconstruction. Therefore, I will focus on my work on image segmentation in this report.

2 Image segmentation

2.1 Task and Dataset

We perform my experiments on the dataset of Pascal VOC 2012, which includes about 2913 segmentation annotations of 20 categories in total. We allocate 600 images to the test set and 2313 images to the train set.

For each image pixel, predict the class of the object containing that pixel or 'background' if the pixel does not belong to one of the twenty specified classes. The output from your system should be an indexed image with each pixel index indicating the number of the inferred class (1-20) or zero, indicating background.

2.2 Data preprocess

As images of VOC 2012 have different size, we should transform them into the same size firstly. However, we can't resize images and annotations directly, because images are annotated pixel by pixel and the mapping from the original image to the annotation will be changed by resizing. Therefore, we perform a random crop of 256x256 to the original image and the annotation at one time and guarantee their crop areas are same. In addition, we also normalize images with mean=(0.485, 0.456, 0.406) and std=(0.229, 0.224, 0.225).

2.3 Fully Convolutional Network

FCN is a basic network for segmentation task. FCN consists of a downsample network and an upsample network. The downsample network is used to extract features from original images and the upsample network is used to map the features to original size. Therefore, we want to explore the influence of different downsample networks and upsample networks.

2.3.1 Downsample

VGG is a classic CNN, which consists of a series of convolutional layers with 3x3 kernel and maxpool layers. VGG-16 with 13 convolutional layers and VGG-19 with 16 convolutional layers are chosen to serve as our downsample networks.

2.3.2 Upsample

Transposed convolution and bilinear interpation are two main upsample methods. The transposed convolutional layer uses a dense kernel to enlarge the matrix, which is similar with the convolutional layer. And bilinear interpation is a mathematical method to upsample, by which the values of unknowed points can be calculated by their neighborhoods.

2.4 Evaluation index

- mean accuracy: the average accuracy of 20 categories and background
- mean iou: the average iou of 20 categories and background

2.5 Experiment

We perform three different experiments:

1. VGG-16 + bilinear interpolation
2. VGG-19 + bilinear interpolation
3. VGG-19 + transposed convolution

2.5.1 Total

	MAP	MIOU
VGG16+interpolation	0.645	0.411
VGG19+interpolation	0.693	0.437
VGG19+transposed convolution	0.641	0.485

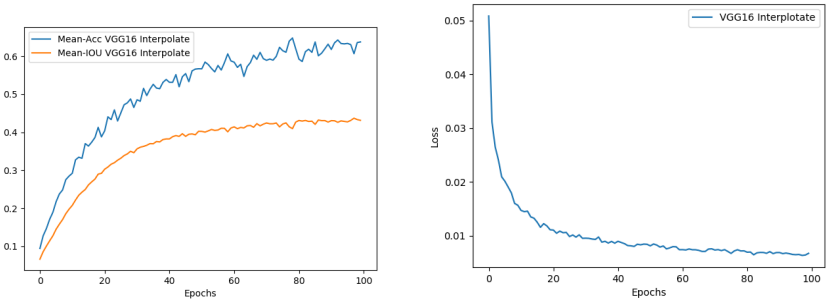

Figure 1: VGG-16 + bilinear interpolation, optimizer:SGD, learning rate=1e-3

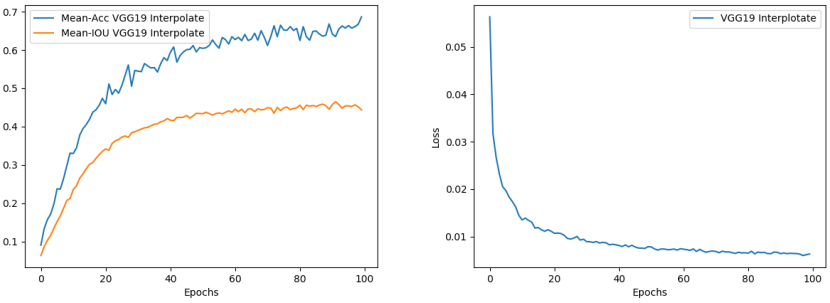

Figure 2: VGG-19 + bilinear interpolation, optimizer:SGD, learning rate=1e-3

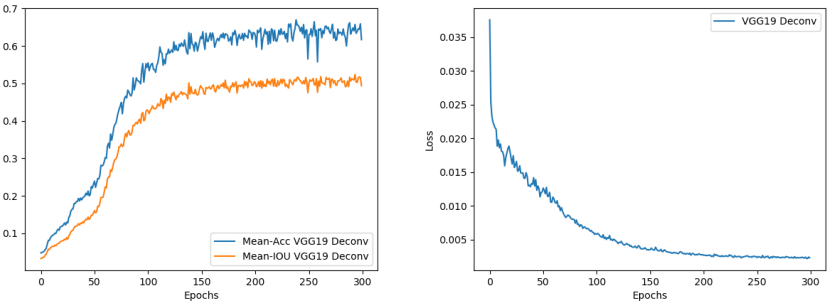

Figure 3: VGG-19 + transposed convolution, optimizer:SGD, learning rate=5e-2

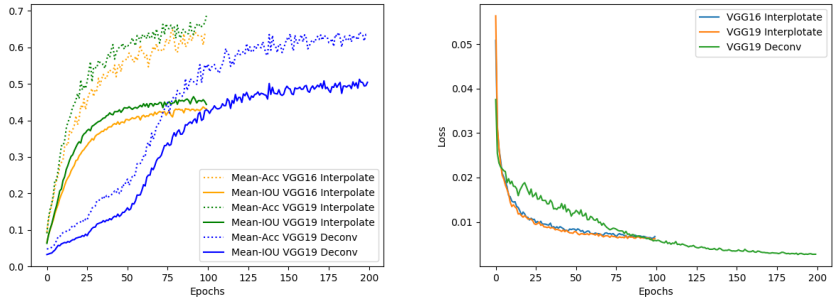

Figure 4: The total result

2.6 Analysis

VGG-19 performs better than VGG-16. VGG-19 has more convoluational layers and is a deeper network than VGG-16, which means VGG-19 has a larger receptive filed, which contributes to extract more efficient features in the segmentation task.

VGG19+transposed convolution performs better than VGG19+interpolation, which is reflected by Mean-IOU and Figure 5. The main difference between bilinear interpolation and transposed convolution is that transposed convolutional layers have far more parameters than bilinear interpolated layers and they can be trained as well. Therefore, transposed convolutional layers can adjust along with VGG19 and learn more details of segmentation. The negative effect is longer training time and larger model.

However, why is VGG19-interpolate has a smaller mean IoU but a larger mean accuracy? Consider the formula of IoU and Acc: $ACC = TP/(TP+FN)$ and $IoU = TP/(TP+FN+FP)$. And $TP+FN$ is the annotation, which can be considered as const. See Figure 5, and observe that VGG19-interpolate intends to generate more rough segamentation with more pixels and larger area, which may result in largger TP and larger FP. Thus, Acc increases. And if the increase rate of FP is larger than TP, then IoU decreases. From the display of Figure 5, mean IoU is a more reasonable index obviously.

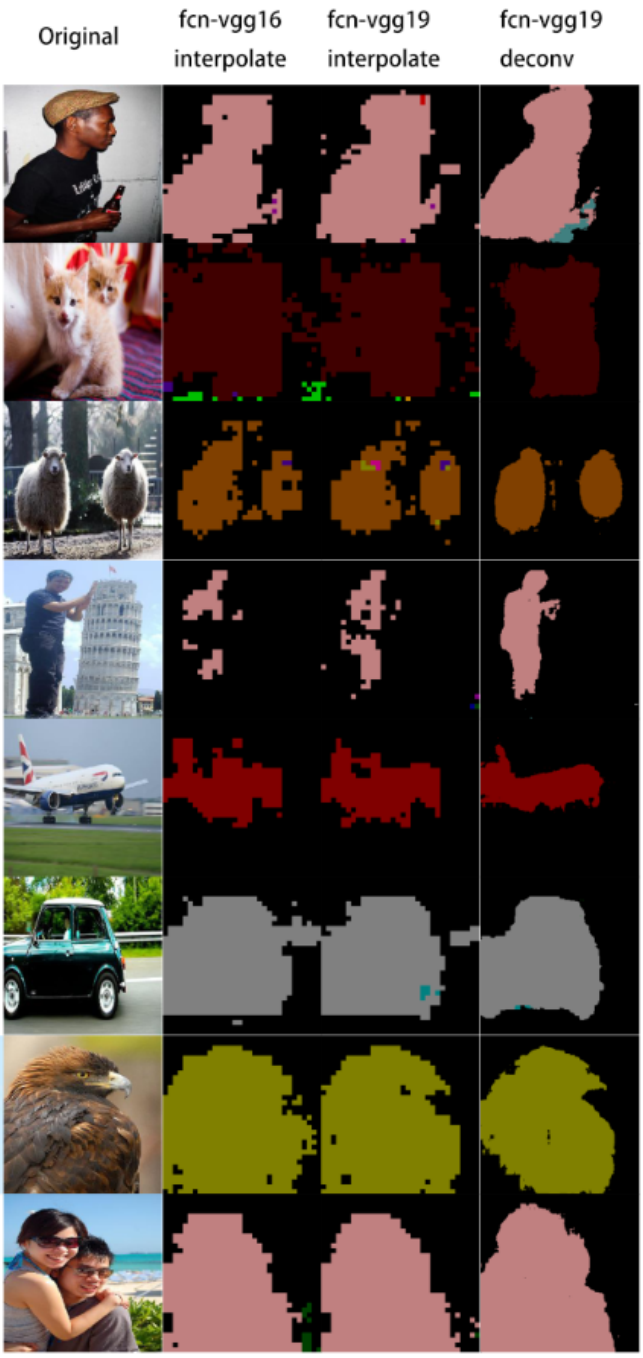

Figure 5: Display of our results

3 Multi-category classification

4 Image reconstruction

5 My problems and experience

5.1 Pre-trained model

I intended to use pretrained model firstly and torchvision provides pretrained VGG . However, I don't know the VGG's structure at all and the pretrained model really confused me. I spent several hours to embed the pretrained VGG into my FCN but it doesn't work.

Then, I gave up and decided to train my VGG directly. However, the network couldn't converge at all because of too small dataset with only about 3000 images and labels. Although it was a failed attempt, but I understood the VGG after building network by myself and realized the problem when using the pretrained model is that the featuremap used to upsample should be the output of maxpool layers while not the output of convolutional layers.

The pretrained VGG could be embedded but another problem raised up: whether should the parameters of VGG be updated? Updating VGG's parameters will take up a lot of computing resources and slow down the convergence. However, if VGG didn't update, the network could learn effectively about new inputs, and the network will even keep frozen if bilinear interpolation was adopted because interpolated layers also have no parameters to update. My solution is to fix the pretrained VGG and add several updatable convolutional layers behind it, which is proved to be feasible to solve the dilemma.

My experience is that before using the pretrained model, build the network by hands step by step to understand its structure, or you will waste lots of time on debugging. And you can modified the pretrained model to make it more suitable for your task.

5.2 How to fit various input sizes?

The sizes of input images are different and an ideal segmentation network is able to accept input images with arbitrary sizes. It's easy to enlarge a matrix to an arbitrary size by interpolation, but something wrong may occure if transposed convolution is used. Consider an FCN that downsamples by maxpool in VGG, and upsamples by transposed convolution.

$$output = (input - 1) * stride + output\_padding - 2 * padding + kernel\_size$$

For example, the size of 32 will be downsampled to 16 by maxpool, and the layer ConvTranspose2d(stride=2, output_padding=1, padding=1, kernel_size=3) can upsample 16 to 32. Now consider the size of 33, which will downsampel to 16 and upsample to 32. It will result in error when merging features from different

layers by matrix addition, which is a significant operation in FCN, because matrix additon requires two matrixs have the same size.

I didn't solve the problem truly but bypass it by limiting the input as 256x256. The training set and validation set will be cropped randomly to 256x256 before entering the network. And the test images will be resized to 256x256 firstly and the output segmentation label is also 256x256. The problem should be fixed in practice but I think it OK in this experiment.

5.3 Dataset

Expand the training set firstly. In VOC 2012 Segmentation, the original $train : val \approx 1 : 1$, so I re-split the dataset as $train : val \approx 4 : 1$ to add training samples. In addition, I perform a random crop on the training images, which helps to increase diversity of training sample.

What's more, the validation set is neccessary. Mean accuracy and mean IoU on validation along with epochs is useful to determine whether training is adequate and advoid overfitting.

5.4 Framework is important

Argparse is helpful to build a framework. If I have multi networks and need to switch between them, a framework is neccessary to decrease redundant codes and advoid modifying another network when debugging.

5.5 Output images directly

Sometimes the evaluation index is confused. For example, compared with fcn-vgg19-interpolation, fcn-vgg19-deconv has smaller mean accuracy but largger mean IoU. It's hard to judge which one is better. However, if output the segmentation results such as Figure 5, the answer is obvious.

5.6 torchvision is useful to process images

In VOC2012, original images are jpg, and labels are png. The torchvision provides functions to transform between different formats including jpg, png and tensor.