Deep Learning Project

Liyuan Liu

October 21, 2018

1 Introduction

In this project, we use semantic segmentation to segment target in the video stream. Each pixel is labeled. Thus fully convolutional network is perfect to perform this task. The input image is first downsized by convolutional operation, and then by using bilinear upsampling method, the downsized images are upsampled to the original input image size. With the help of mask of the images, the right labels are made for each pixel. Then it is a standard classification problem.

2 The model architecture

The model architecture looks like below,

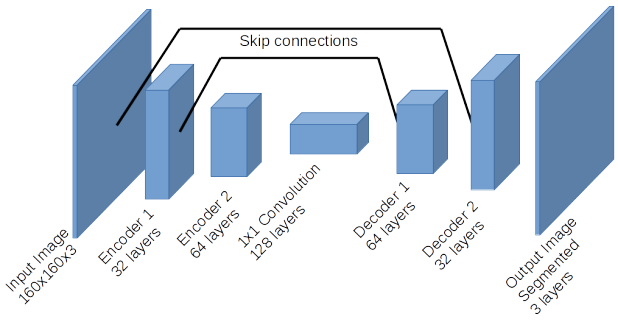

Figure 1: The model architecture

We can see that the input images are first downsized by the convolution operators. This downsized part of the network played as the image extraction operation. It can encode the image information. For the upsampling part of the network, it maded use of the previous part and did the upsampling to match the orignal input images so that we can use cross entropy loss function to measure

the performance of this network.

The skip connections are used to retain the information during the down-sampling which helps improve the accuracy of pixel-level classification. The final iou is 58.86% and the final score is 0.47.

I used 2 layers to encode information from the images. The first layer is 32 output layers and the kernel size is 3. The second layer has 64 output layers and the kernel size is also 3. Then there is a $1x1$ convolutional layer with 128 output layers follows. There are also 2 layers of decoder. The first has 64 output layers. The second has 32 output layers. All the activation function is relu which is simple and effective.

I have also tried other network architectures, such as three layers of down-sizing and three layers upsampling. But the final iou is only 47% and the final score is less than 0.4. After a series of experiments, I finally choose the network architecture above.

3 Hyperparameters

I focus on demonstrating 3 hyperparameters which are the number of epochs, the learning rate and the batch size.

For the number of epochs, I tried 10 and 18 epochs. For 10 epochs, the final iou is 40.5% and the final score is 0.37. It means that the model is underfitting. Thus then I increased the number of epochs to 18. Then the final iou is 58.86% and the final score is 0.47.

For the learning rate, I have tried 0.1, 0.001 and 0.0001. For 0.1, the training loss decreased faster at first, but it can not go down any further in the following epochs. And for 0.0001, the training loss went down smoothly, but a little slow. Then I chose 0.001, it seemed that the training loss went down at a proper speed and did not stopped decreasing.

For the batch size, I tried 32 and 128. For 128, the weights updating frequency is slower than 32, but the performance is just the same as 32. Thus finally I chose the batch size of 32.

4 One by One Convolution V.S. Fully connected layer

One by one convolution is nothing but the convolution operation with the kernel size of 1. Thus actually it does not change the width and height of the output if the stride is 1 too. But it can change the depth of the output. Thus it can significantly influence the size of the output. For example when we want to reduce the number of the weight parameters and want to retain the spatial information, we can use one by one convolution to reduce the number of channels of the

output. Because it can maintaiin the spatial information, in the fully convolutional network, one by one convolution is a perfect tool to reach out destination.

For the fully connected layer, it is similar to one by one convolution. But it can not maintain the spatial information. In the image level classification problem, we always use fully connected layers to decrease the image dimension from 2 to 1 and then the 1 dimensional vector can be used to do classification.

5 Fully convolutional Network

In the fully convolutional network, there are two parts which are the encoder and the decoder. With encoder we can extract the information from the input images. During the extraction process, the height and the width are reduced and the depth is increased. And since the problem is a pixel-level classification problem, we need to reconstruct the size of the input images. The decoder part is reponsible for this task. There are many methods that we can use to do upsampling, such as transpose convolutional layer and bilinear upsampling. Also we use skip connections to make the upsampling more robust.

The problem that may rise in this process is intensive computing. It needs a lot of training samples and powerful enough computers to get good results.

6 Inference for other objects

Can we use this model trained for the person to make prediction for following other objects such as dogs and cats? The answer is obviously no. Because the weights of the model are trained on the input images of people. It can not extend to infer dogs and cats with those weights. To make inference for other objects, we need to collect a lot of samples of those objects and train the network from scratch. That is the limitation of neural network.

7 Future enhancement

To further improve the accuracy, we can collect more data when there is no target in the scene and when there is target but the target is far away and very small in the image surrounded by many other people.

Besides, we can also try to make the network more complex when there are much more data for training.