Udacity

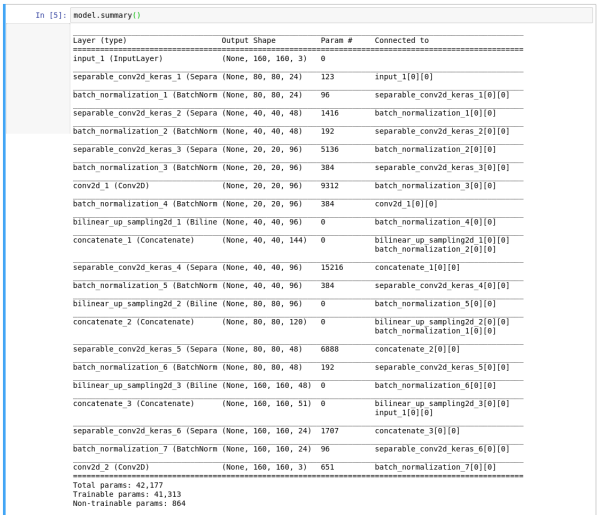
Christian Leininger

Supervisor:
Advisor:

1 The write-up conveys the an understanding of the network architecture.

The model has 3 Encoder at the beginning and 3 Decoder at the end between them is a batch norm layer and the an output layer for classification. Experiments shown that this size is not under or over fitting to the given training data. Parameters:
filtersize = 24 depends on the depth of the layer
strides = 2 in Encoder and 1 every other layer
kernelsize = 1
activation function = softmax for classification

1.1 Network



```
In [5]: model.summary()
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 160, 160, 3)	0	
separable_conv2d_keras_1 (Separa	(None, 80, 80, 24)	123	input_1[0][0]
batch_normalization_1 (BatchNorm	(None, 80, 80, 24)	96	separable_conv2d_keras_1[0][0]
separable_conv2d_keras_2 (Separa	(None, 40, 40, 48)	1416	batch_normalization_1[0][0]
batch_normalization_2 (BatchNorm	(None, 40, 40, 48)	192	separable_conv2d_keras_2[0][0]
separable_conv2d_keras_3 (Separa	(None, 20, 20, 96)	5136	batch_normalization_2[0][0]
batch_normalization_3 (BatchNorm	(None, 20, 20, 96)	384	separable_conv2d_keras_3[0][0]
conv2d_1 (Conv2D)	(None, 20, 20, 96)	9312	batch_normalization_3[0][0]
batch_normalization_4 (BatchNorm	(None, 20, 20, 96)	384	conv2d_1[0][0]
bilinear_up_sampling2d_1 (Biline	(None, 40, 40, 96)	0	batch_normalization_4[0][0]
concatenate_1 (Concatenate)	(None, 40, 40, 144)	0	bilinear_up_sampling2d_1[0][0] batch_normalization_2[0][0]
separable_conv2d_keras_4 (Separa	(None, 40, 40, 96)	15216	concatenate_1[0][0]
batch_normalization_5 (BatchNorm	(None, 40, 40, 96)	384	separable_conv2d_keras_4[0][0]
bilinear_up_sampling2d_2 (Biline	(None, 80, 80, 96)	0	batch_normalization_5[0][0]
concatenate_2 (Concatenate)	(None, 80, 80, 120)	0	bilinear_up_sampling2d_2[0][0] batch_normalization_1[0][0]
separable_conv2d_keras_5 (Separa	(None, 80, 80, 48)	6888	concatenate_2[0][0]
batch_normalization_6 (BatchNorm	(None, 80, 80, 48)	192	separable_conv2d_keras_5[0][0]
bilinear_up_sampling2d_3 (Biline	(None, 160, 160, 48)	0	batch_normalization_6[0][0]
concatenate_3 (Concatenate)	(None, 160, 160, 51)	0	bilinear_up_sampling2d_3[0][0] input_1[0][0]
separable_conv2d_keras_6 (Separa	(None, 160, 160, 24)	1707	concatenate_3[0][0]
batch_normalization_7 (BatchNorm	(None, 160, 160, 24)	96	separable_conv2d_keras_6[0][0]
conv2d_2 (Conv2D)	(None, 160, 160, 3)	651	batch_normalization_7[0][0]

```
Total params: 42,177
Trainable params: 41,313
Non-trainable params: 864
```

Figure 1: The summery of the model

2 Hyperparameter

The hyperparameter search was performed by a grid search
Network: 3 Encoder 1 , 1x1 conv, 3 Decoder
Epoch: 100 after that no significant improvement was detected
Learning-rate: 0.004 the best results of some similar learning rates
Batch-size: amount of Images after the network computes the loss and performs back propagation on the network weights to reduce the loss.
steps per epoch: 32 amount of batches each epoch are used
validation steps: 50 check the generalization of the network

3 understanding

3.1 Batch Normalization layer

Here the Input is getting normalized by using the mean and varinace of the batch input. Advantages are
- has been shown work as well as the dropout technique for regularization
- allows higher learning rates to speed up training

3.2 connected layer

3.3 1by1 convolutions

This layer is used when the with and height of an layer want to be preserved and the amount of the features should to be reduced. I used the layer given in Keras Framework SeparableConv2DKeras(reduces parameter to the standard) with the parameters filters, kernel, strides, padding and activation function. By seting strides to 1 we obtain the 1x1.

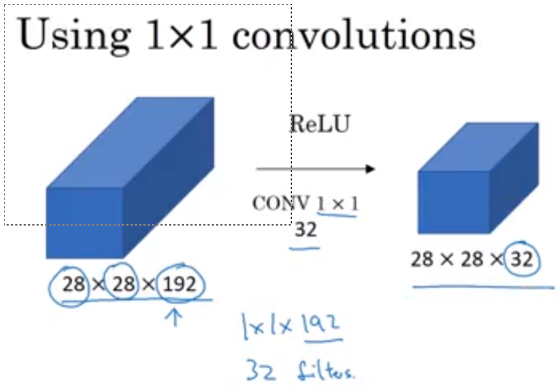


Figure 2: Example of 1x1 convolution

In the Example above 1x1 convolution with 32 filter was used to get from 192 features to the 32. Each of the 32 filter has size of 1x1x192 the 192 is from the Input feature.

4 Encoding and Decoding images

FCN Encoder:
The Encoder reduces the with and height of the Input to extract features out of it
FCN Decoder:
After we reduced the with and height of the Input the layers need to up sample the layers to the size of the Input. In order to archive this we use a method called Bilinear Upsampling. This is a resampling technique that utilizes the weighted average of four nearest known pixels, located diagonally to a given pixel, to estimate a new pixel intensity value.
With skip-connetion we connect some layer from the Encoder to Decoder.
The Encoder-decoder models face two main problems. The one is structural stereotype which is receptive fields imbalance rooted in this kind of frameworks. The other is insufficient learning that deeper neural networks tend to encounter the notorious problem of vanishing gradients. Structural stereotype leads to unfair learning and inhomogeneous reasoning.

5 Display a solid understanding of the limitations to the neural network with the given data chosen for various follow-me scenarios

This model would not be able to follow other objects since it was trained to follow this person. In order that the model is able to follow other objects it needs to be trained on data with that object and probability the Network needs to changed as well and also other Hyperparameters.

6 Future Enhancements

First idea:
Bayesian SegNet to get an measure of the uncertainty of the models prediction Pre train the encoder or use trainsfer learning and take an already pretrained network like vgg16

References