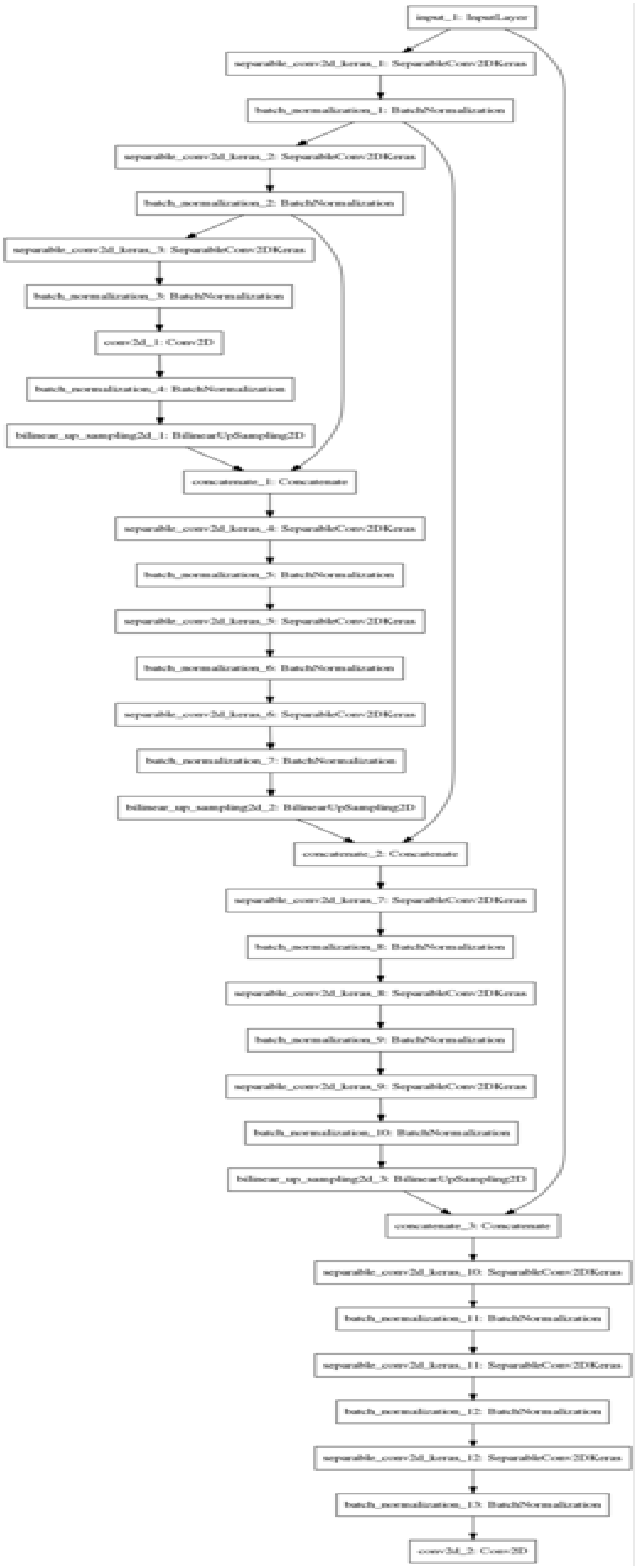
Andrew Truong

Robotics Nanodegree Deep Learning Project- Follow Me

Project Description

The project focuses on using a fully convolutional network (FCN) to perform semantic segmentation. The project uses data from the quadcopter simulation where humans and a hero are spawned. The quadcopter patrols on a path and captures various images. The network then uses these images to segment each image into one of three categories: person, hero, or background. The quadcopter can use this data to follow to hero in real time in the simulation.



The Fully Convolutional Network

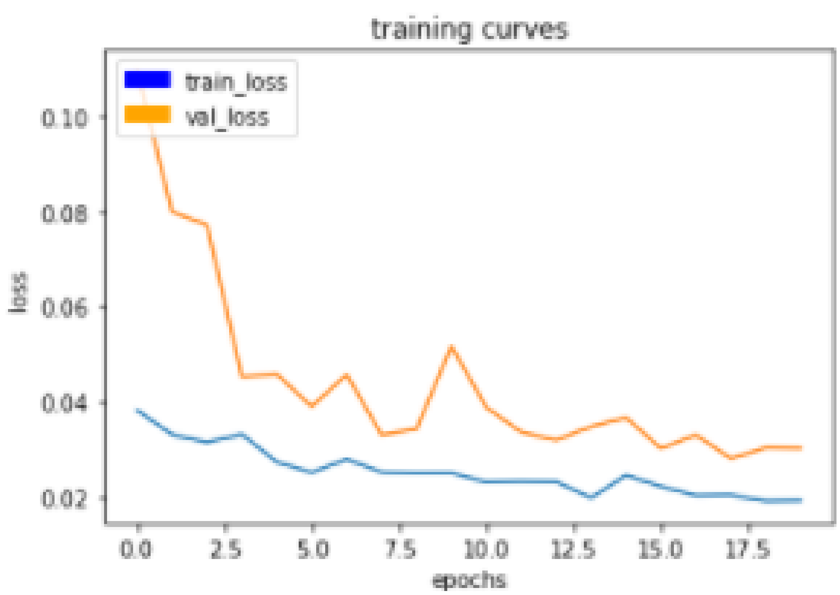
The FCN uses separable convolution layers to improve model performance and decrease processing time. The input layer uses ReLU to accelerate the convergence of stochastic gradient descent inexpensively. The network uses three encoder blocks of size 32, 32, and 64 with strides = 2. The 1x1 convolution layer is of size 64 with strides = 1. The 1x1 layer reduces the dimensionality of the previous layers while increasing the depth to retain the necessary information for segmentation. Fully connected layer have the same dimensionality as previous layers. The pooling method takes in max numbers and average numbers to keep the most important information while reducing the dimensionality. Low value information about the image is loss through pooling and downsampling, so the image resolution decreases. The solve this issue, the model uses skip connections. Information from previous layers are used to sum the layers togethers and create a model that can accurately segment images. This generates a deeper network without stacking more layers. Spatial information, HxD, and coarseness remains the same. The encoder block goes through dimensionality reduction to reduce the number of inputs for the decoder to reconstruct. The encoder takes input images from the data set and creates high dimensional feature vectors. Upsampling takes the downsampled data builds the images with the same spatial information through unpooling. The decoder takes the high dimensional feature vector and creates the semantic segmentation mask. The decoder blocks are sized 64, 64, and 32. Training the model is computationally intensive, so the AWS cloud computing service was used with the px2.large.

Hyperparameters

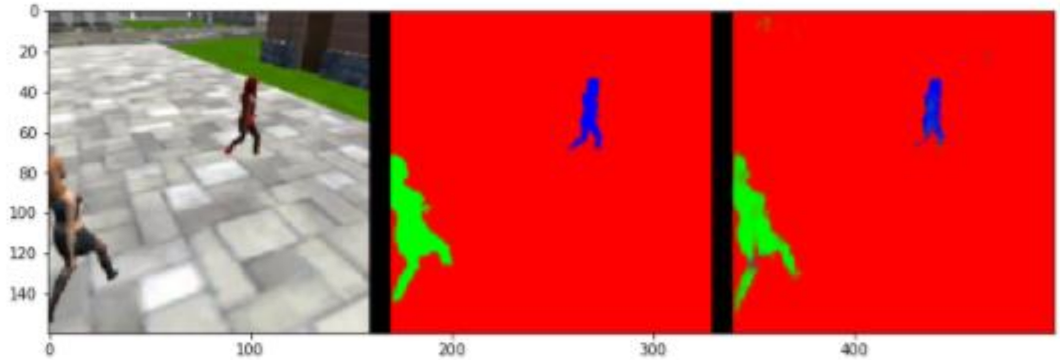
The hyperparameters were tuned with brute force. When the learning rate was decreased, model performance increased, and training was more reliable. However, If the learning rate was too low, the model would take too long to train. Increasing the steps per epoch improved the model performance. More steps per epoch also increased the training time. The hyperparameters used for this model are:

```
learning_rate = 0.005
batch_size = 64
num_epochs = 20
steps_per_epoch = 50
validation_steps = 50
workers = 2
```

The network training curves are used to minimize training and validation loss. Increasing the steps per epoch or the number of epochs was shown to improve the performance of the model. The final training curve resulted in a training loss of 0.0191 and a validation loss of 0.0301.



The model then performed semantic segmentation on images to predict objects in the simulation. The performance of the predications was based on the Intersection over Union (IoU) method. The IoU score was 0.557 for this model. The final score includes the weights based on false positive and negatives and true positives and negatives. The final score of the model is 0.414.



Future Enhancement

The model could be improved to detect more objects. The drone could use this information for object avoidance when the hero moves in a pathway with many obstacles. The model currently does not segment objects other than humans. This model would not work well for other objects, because the model was only trained on humans. To detect objects such as dogs, cats, cars, etc, the model will need to be trained on data in the simulator with corresponding mask images. More training data would need to be obtain for the network to detect and classify objects.

References

http://cs231n.github.io/neural-networks-1/

https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

http://iamaaditya.github.io/2016/03/one-by-one-convolution/

https://datascience.stackexchange.com/questions/12830/how-are-1x1-convolutions-the-same-as-a-fully-connected-layer

https://stats.stackexchange.com/questions/194142/what-does-1x1-convolution-mean-in-a-neural-network

https://www.quora.com/What-is-an-Encoder-Decoder-in-Deep-Learning/answer/Christian-Baumgartner-3?srid=ison

https://courses.cs.washington.edu/courses/cse576/17sp/notes/Sachin_Talk.pdf

https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html

https://hackernoon.com/autoencoders-deep-learning-bits-1-11731e200694