CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 1

Read

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 2

Read

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 3

Read

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 4

Read

Solution: Complete ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 5

Compute the receptive field size at the input to the global average pooling layer for ResNet 50.

Solution: 164×164 ■

Proof: Looking at the architecture of Resnet-50 we see that there are a total of six downsamplings, each of which alter receptive field. Four of the downsamplings take place in the body, and two in the head. The body downsamplings are $3 \times 3/2$ and the head downsamplings are $3 \times 3/2$ and $7 \times 7/2$. We can then compute receptive field by starting with 1, multiplying by stride and adding filter size minus one.

$$\text{conv5} = 1 \tag{1}$$
$$\text{conv4} = \big(\text{conv5} + 2\big) * 2 = 6 \tag{2}$$
$$\text{conv3} = \big(\text{conv4} + 2\big) * 2 = 16 \tag{3}$$
$$\text{conv2} = \big(\text{conv3} + 2\big) * 2 = 36 \tag{4}$$
$$\text{max pool} = \big(\text{conv2} + 2\big) * 2 = 76 \tag{5}$$
$$\text{conv1} = \big(\text{max pool} + 6\big) * 2 = 164 \tag{6}$$
$$\tag{7}$$

This result seems resonable, as it captures enough of the original image to resolve an object. It seems difficult to find solutions to sample calculations for full networks, **additional problems with verifiable solutions would be helpful**.

□

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 6

(a) How does the accuracy of the half wide version copare to the original version?

Solution: The accuracy at the final epoch for the full and half versions are

$$\text{Full} = 91.03\% \qquad \text{Half} = 89.31\% \tag{8}$$

■

(b) How long does an epoch of training take for both versions?

Solution: The times for epoch 3 for both models (trained on GTX970) are

$$\text{Full} = 59s \qquad \text{Half} = 36s \tag{9}$$

■

(c) Approximate how feature map memory, filter memory, and compute change between the full and half width versions.

Solution:

$$\text{Feature map memory} = \frac{1}{2} \tag{10}$$

$$\text{Filter memory} = \frac{1}{4} \tag{11}$$

$$\text{Compute} = \frac{1}{4} \tag{12}$$

■

Proof: In our approximation we can look primarily at how convolutional layers will be affected, as such layers comprise the bulk of the network. In this case we know that

$$\text{Filter Memory} = N_o * N_i * F_r * F_c + N_o \tag{13}$$

$$\text{Feature Map Memory} = N_{o/i} * L_r * L_c \tag{14}$$

$$\tag{15}$$

As such we expect to see a $1/2$ reduction in feature map memory, which makes sense given that the half wide version will store half as many feature maps. We also expect a $1/4$ reduction in filter coefficient memory as we are now mapping across $N_i = 1/2$ and $N_o = 1/2$ for a combined $1/4$ reduction.

For compute of 2D convolution we have

$$\text{Compute Memory} = N_o * N_i * L_r * L_c * F_r * F_c \tag{16}$$

which again contributes to a quarter reduction over $N_o * N_i$.

□

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 7

Solution: The network design will be based on Inception v4. To inform our choice of network architecture we can use the given hint for an expected input global average pooling and think backwards from here. Obviously for CIFAR-10 we will have 10 classes, so we need to choose N_i to global average pooling somewhere above 10, call it 30. We will aim for 7×7 spatial input to average pooling.

Again, recall that downsampling is given by

$$M = \frac{L - F + 2P}{S} + 1 \tag{17}$$

Putting this together, our architecture looks like

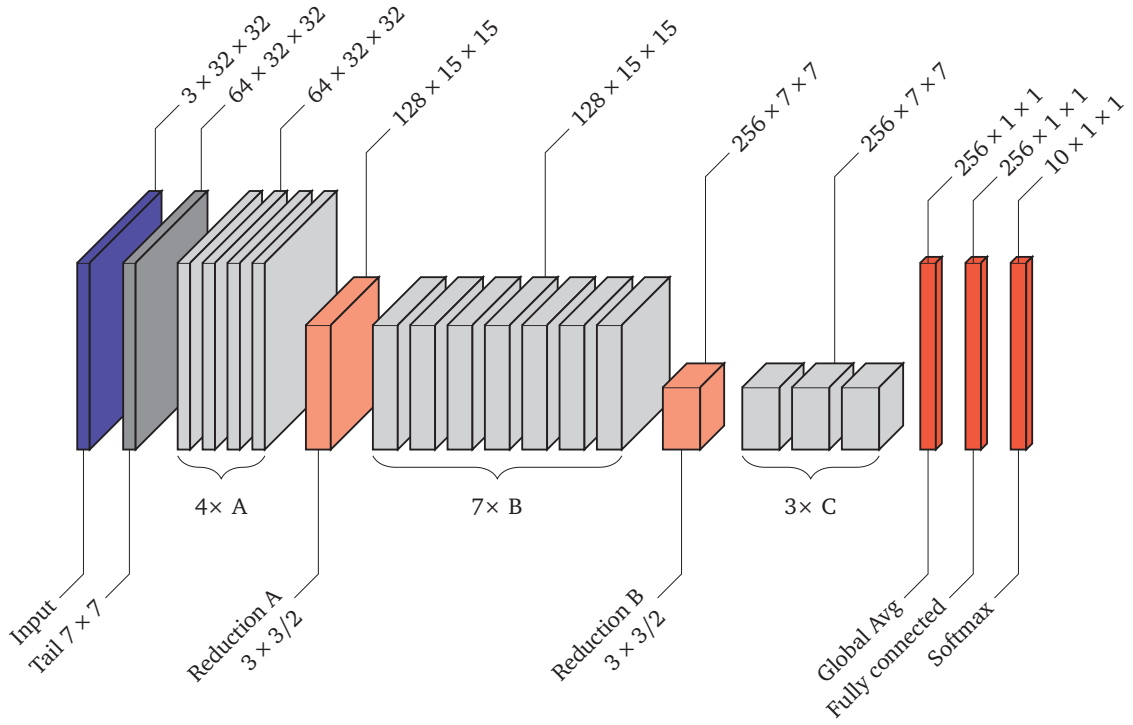

Figure 1. Possible Inception v4 architecture. Input dimensions are given

The max pooling operation usually included in the tail was dropped to avoid excessive down sampling. It may have been possible make adjustments to accomodate the stride by two, or the stride could have been left out from that tail component.

To calculate the receptive field size we can traverse backwards through the reduction blocks, both of which are $3 \times 3/2$.

$$L = \Big(\big(1+2\big) * 2 + 2\Big) * 2 \tag{18}$$
$$= 16 \tag{19}$$

So we capture about a quarter of the total image area in a given receptive field.

Now we need to compute memory. We can illustrate this on the same network diagram, with feature map memory on top and filter coefficient memory on bottom. Recall that we will be computing

$$\text{Filter Memory} = N_o * N_i * F_r * F_c + N_o \tag{20}$$
$$\text{Feature Map Memory} = N_{o/i} * L_r * L_c \tag{21}$$
$$\tag{22}$$

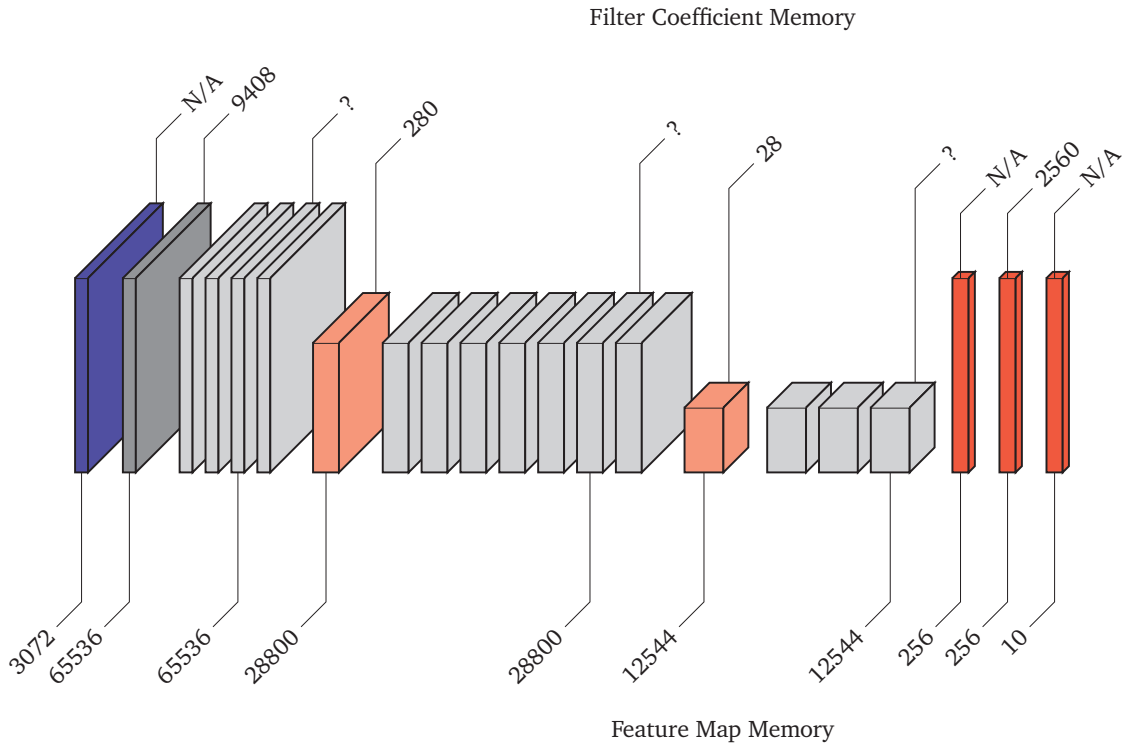

Figure 2. Possible Inception v4 architecture. Input dimensions are given

I ran out of time to compute the filter coefficient size for the inception layers, though one can see how the convolution layers that comprise these blocks will see a rapid rise in feature map count as we get deeper in the network. The network will be compute heavy near the tail, where we benefit from having feature maps reside in memory from layer to layer. Deeper in the network we will have more memory movement.

■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 8

Solution: Attempted. Inception v4 ended up being particularly difficult to implement. This revealed the importance of building small but robust blocks from which the entire network will eventually be composed. It also revealed the potential difficulty of debugging design errors, as Tensorflow exceptions were quite lengthy.

Ran into an issue where Tensorflow would raise CUDA ERROR UNKNOWN when trying to run on local GPU. Only solution was to reboot. Finally, I experimented with implementing GPU memory restrictions to improve training speed. My system uses two GTX970s which suffer a memory issue where 0.5 of the total 4GB runs significantly slower. Resticting memory usage and training on the secondary GPU that was not rendering X11 seemed to improve performance. ■

CS 6301.503 Spring 2019 Scott C. Waggener (scw180000)

Homework 6 (Design) Problem 9

Solution: I resorted to trying variations of and ResNext after Inception proved impractical to implement, however I was not able to exceed the accuracy of Resnet for CIFAR-10. ■