Journal Report 8
10/28/19-11/1/19
Odin Woitek
Computer Systems Research Lab
Period 1, White

Daily Log

Monday October 28

Since the image patching is random, my partner wanted me to upload the data to github so we could train our model on the same dataset. Unfortunately, I ran into the problem that our data, which is composed of many images, exceeded the maximum storage space allowed on github of 100 MB. I tried to find a way around this through using git large file storage, but I couldn't get it to work. I ended up just uploading the folder to google drive and sharing it with him, but this whole process made me think a bit more about the amount of data we are using, which I think may end up being unnecessarily large.

Tuesday October 29

I researched a bit about ResNets and their architecture. I began writing a method for an identity block, which functions as the skip connection that allows the gradient to be passed to a deeper level of the network, thereby avoiding the vanishing gradient problem. I will make a method for an identity block and a method for a convolution block. Once that is done, it will be easy to build the model by calling these methods to form the blocks that create the model.

Thursday October 31

I worked on setting up Zoidberg so I can run my model quickly once I finish building it. A few weeks ago, I looked into running the model on GPU but found out I couldn't do it on my laptop, so I am very glad this opportunity exists.

Wednesday November 6

I wrote the convolution block, which is similar to the identity block except it creates a shortcut path as well as the main path and combines the two paths at the end. I also took the ResNet 50 layer from the tutorial I was using to use as my baseline network.

Thursday November 7

I altered the ResNet 50 layer network from the tutorial to function with my data and ran it. I had to change my input to have the depth at the end in order to avoid having a negative dimension resulting from my max pooling layer. I ended up being able to have the model compile and begin running, but it is very very slow and has low accuracy. I'm having trouble understanding how the model works, and the paper we are referencing doesn't even us a ResNet, so I'm going to go back to something simpler with the knowledge that I gained from what I have tried to do here.

Timeline

Date	Goal	Met
October 7-11	Load input and output data into training and testing sets	Yes, Used sci-kit learn to split data into training, testing, and validation sets.
October 14-25	Run and evaluate model	Yes, using the architecture of the MNIST tutorial model, I got an accuracy of about .27 and a loss of about 2.73
October 28 - November 8	Continue research on ResNet and implement a 50 layer ResNet	Yes, but it ended up being worse than the network I had before
November 11-15	Modify previous network	
November 18-22	Improve accuracy to 0.4	

Reflection

This week, my goal was to implement the ResNet. I started working on it, but I had to spend a lot of time researching ResNets before I understood how they work and I also ended up spending time working on other aspects of my project like dealing with our input data, and acknowledging that it may be unnecessarily large. I did set up Zoidberg, so when I finish the 50 layer ResNet, I can run it quickly. I made a a modified version of a ResNet 50 from a tutorial maintaining the overall model architecture both in the way the blocks are laid out and the layers within the blocks themselves. I came across the same subtracting 3 from 1 issue from a few weeks back, which I fixed by moving the input layer with the dimension of 1 to the last layer instead of the first, so the padding layers wouldn't leave this with a negative number of dimensions in the first layer. I ran the network on my computer just to see what it does, and I was very disappointed in how slowly it ran and how inaccurate it was (although the accuracy may improve if it can run faster). Next week, I will go back to work on the previous model I had, but I will also run both networks with Zoidberg, since that will give me a better idea of how they stack up against each other considering both speed and accuracy if the ResNet's problem truly is just its speed.

Here is the identity block:

```
def identity_block(X, f, filters, stage, block):
    #Name base
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'

    #extract filters
    F1, F2, F3 = filters

    # Save the input value
    X_shortcut = X
```

```
#Three components:
X = Convolution2D(filters = F1, kernel_size = (1,1),strides=(1,1),padding='vali
        name=conv_name_base+'2a',kernel_initializer=glorot_uniform(seed=0))(
X = BatchNormalization(axis=3, name = bn_name_base+'2a')(X)
X = Activation('relu')(X)

X = Convolution2D(filters = F2, kernel_size = (f,f),strides=(1,1),padding='same
        name=conv_name_base+'2b',kernel_initializer=glorot_uniform(seed=0))(
X = BatchNormalization(axis=3, name = bn_name_base+'2b')(X)
X = Activation('relu')(X)

X = Convolution2D(filters = F3, kernel_size = (1,1),strides=(1,1),padding='vali
        name=conv_name_base+'2c',kernel_initializer=glorot_uniform(seed=0))(
X = BatchNormalization(axis=3, name = bn_name_base+'2c')(X)

#Add shortcut
X = Add()([X, X_shortcut])
X = Activation('relu')(X)

return X
```