**HW4 – Theoretical Questions**

**Task 2 – Activation function. How does changing from ReLU to tanh affect the results?**

The accuracy for the model which used the Relu function was 0.6743 as opposed to 0.6514 in the model which used tanh. This means that our model was more accurate using the Relu function. The limit of tanh is that it saturates for large and small values (the derivative is zero and the algorithm has a difficult time learning). Another advantage of Relu is time and sparsity, the computing time will be much faster than that of tanh function because all the negative values will be zero and thus fewer neurons are firing (this can also be a disadvantage because it is possible that the negative data can give us valuable information and with this activation function we are losing it).

**Task 3 – number of epochs**

In our data, we can see that when we raised the number of epochs from 25 to 40 our accuracy improved a little bit (from 0.6514 to 0.6629). An epoch is a single pass through all training set examples therefore we would expect that a larger number of epochs would give us more accurate results. However, there is a point in which we get to overfitting (when the model overfits the training data and then is less accurate when trying to predict the testing data) so there is a tradeoff (choosing the highest number is not the right way to go).

**Task 4 – What are the advantages of mini-batches vs SGD?**

In batch gradient descent we compute the mean gradient of all the gradients of the training data, in mini batch gradient descent we do the same thing but for smaller batches of the data (not the entire training set). The advantage of this over stochastic gradient descent is that we can get closer to the minimum (in SGD we fluctuate around the minimum but never reach it). In addition, SGD is more sensitive to noise than mini-batch.

**Task 5 – How do batch normalization layers impact our results?**

Batch normalization layers are supposed to improve learning speed and outcomes by normalizing the activations of each layer. When we added the batch normalization layers our accuracy actually went down (0.6171). When we look at our data we see that the images are quite blurry so it is possible that the quality of our input was too low such that improving the model does not improve the results.

**Part 2**

***Task 1:*** *2D CNN.*

Have a look at the model below and answer the following:

- How many layers does it have?
- How many filter in each layer?
- Would the number of parmaters be similar to a fully connected NN?
- Is this specific NN performing regularization?

There are 8 layers in total, 5 hidden convolution layers (which use filters) each one consisting of possible dropout, pooling and batch normalization followed by 3 dense layers. There are 64, 128, 128, 256, 256 filters in layers 1,2,3,4,5 accordingly.

Regarding the number of parameters, we would expect a CNN network to have a lot less parameters than a fully connected network. One of the goals of CNN is to reduce the number of parameters.

We can see that the model uses kernel regularization of l2 so this specific NN is performing regularization.

**Task 2 – what are the results when we train the model with the number of filters cut by half?**

The accuracy of our model with the original number of filters was 0.3428. When we reduced the filters by half the accuracy went down to 0.2457. It is possible that with less filters the network was unable to recognize certain objects and therefore the accuracy was impacted.