HW4 – XRAY

Guy & Dor

Part 1

Task 2:

Change the activation functions to LeakyRelu or tanh or sigmoid. Name the new model new_a_model. Explain how it can affect the model.

Activation function influence models greatly. The activation function determines the output of each neuron given its weighted sum of inputs.

ReLU (Rectified Linear Unit) function is the following function:

$$f(x) = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases}$$

This function is a very useful and commonly used for its fast activation and deactivation of neurons (neurons deactivate when they get the value of 0)

This is also a disadvantage, for its “shutting down” neurons that might be useful or important in some cases. having said that, this function is more commonly used for its fast computation time.

Basic Sigmoid Function:

$$f(x) = \frac{1}{1 + e^{-c_1 \cdot (x - c_2)}}$$

Sigmoid function is also a very useful activation function, the output of the sigmoid is a range between zero and one but there is only a small range where the output is not close to zero or one. Therefore, the function is a good model for a binary switch.

Task 3:

Train the new model using 25 and 40 epochs. What difference does it makes in term of performance?

Lerger number of epochs takes longer time but improves the performance, as we in the results.

Task 4:

Build the model_relu again and run it with a batch size of 32 instead of 64. What are the advantages of the mini-batch vs. SGD?

Both mini-batch and SGD (Stochastic Gradient Descent) are iterative optimization algorithm (Gradient Descent). While SGD takes one example at a time, the mini-batch takes a small batch to calculate the gradient at each step. It is not as slow as batch gradient descent which uses all the training set each iteration, but also smoother than SGD.

SGD is fast, especially in large data sets (comparing to other gradient descent) but the downside is it only get us close to the minimum value and just keep "dancing" around it.

Mini-batch gradient descent is a mix of batch gradient descent and SGD, so it has the advantage of calculating the mean of the gradient and therefore can converge to the minimum value.

Task 4 (2):

Build the new_a_model again and add batch normalization layers. How does it impact your results?

The batch normalization did not made much difference, probably because of the quality of the data.

Part 2

Task 1:

The Neural Network consist of 8 layers. Including 5 convolution layer that use filters, 3 more layer that are fully connected.

Each layer has the following filters: 64, 128, 128, 256, 256 (corresponding to the convolution layers).

We expect the number of parameters in the CNN be much lesser than a fully connected neural network. Because of each convolution layer is not connected to the next layer completely its number of parameters drop dramatically.

This Neural Network uses regularization, we can see that because the convolution layers use kernel regularization of l2.

Task 2:

Results we received:

Filter:64, 128, 128, 256, 256:

Loss: 8.6

Accuracy: 25.1%

Filter:32, 64, 64, 128,128:

Loss: 4.6

Accuracy: 25.7%

These results might happen because the model is overfitted, therefore our model cannot "learn" only memorize.

Because of that our results are poor and reducing the number of filters helped.