Machine Learning in Healthcare

336546

HW4

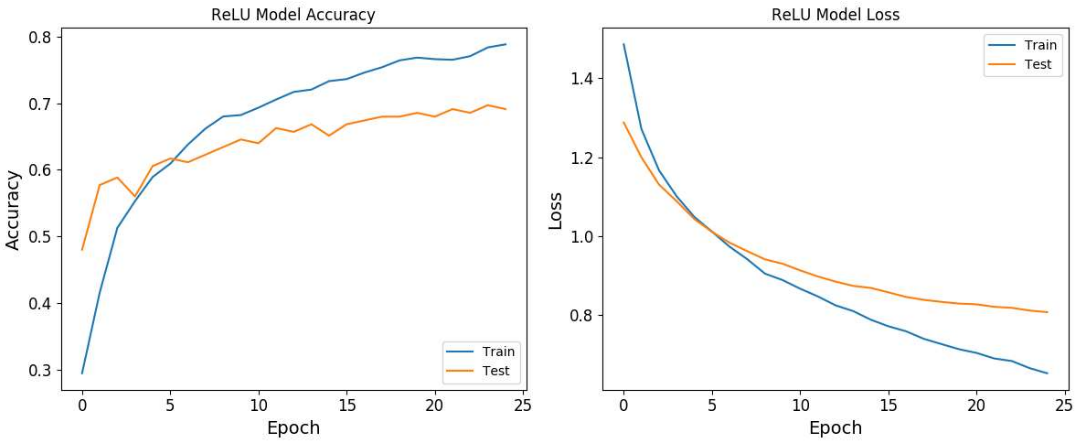
Theoretical Questions

Nathan & Dekel

Part 1: Fully Connected Layers

Task 1: NN with fully connected layers

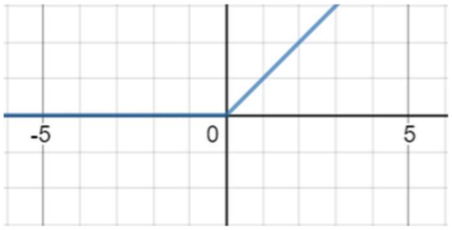
The accuracy and loss over the testing set for the first ReLU model is:



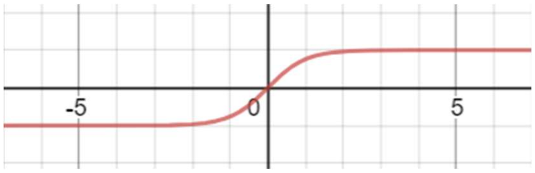
Result of evaluation → Loss =0.8074, Accuracy = 0.6914

Task 2: Activation functions

The activation function determines the output of a neuron given its set of inputs. In our case, ReLU (Rectified Linear Unit) outputs 0 for any non-positive input, and outputs the input for any positive input, as shown by the following graph:



We decided to change the activation function to a tanh activation function (hyperbolic tangent), which maps the input between -1 and 1, where very negative inputs are output as -1 and very positive inputs are output as 1, as shown by the following graph:



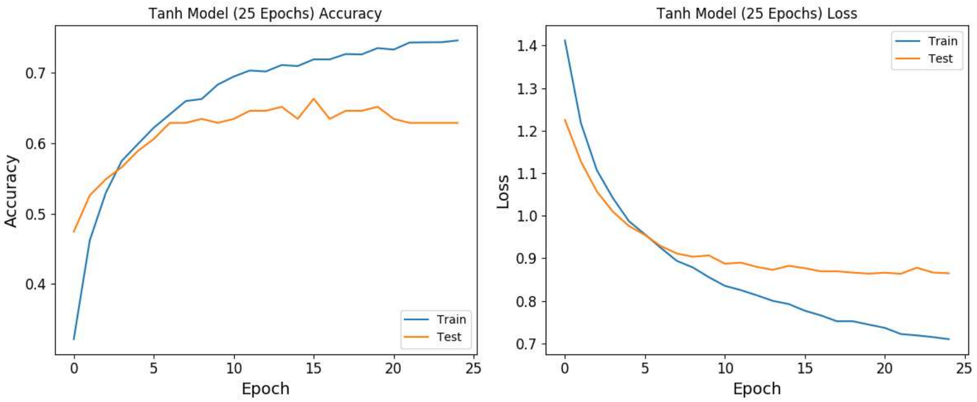
Since our data is normalized, we believe that the tanh activation function can perform as good as (if not better than) the ReLU activation function, since the inputs will not be very positive or negative and therefore will avoid saturation by the tanh activation function. Avoiding saturation means that for two very positive inputs the output will be 1, or -1 for two very negative inputs.

Task 3: Number of epochs

An epoch refers to one cycle through the whole training dataset. As we increase the number of epochs, we reduce the error on the training data, but increasing the number of epochs too much can result in overfitting.

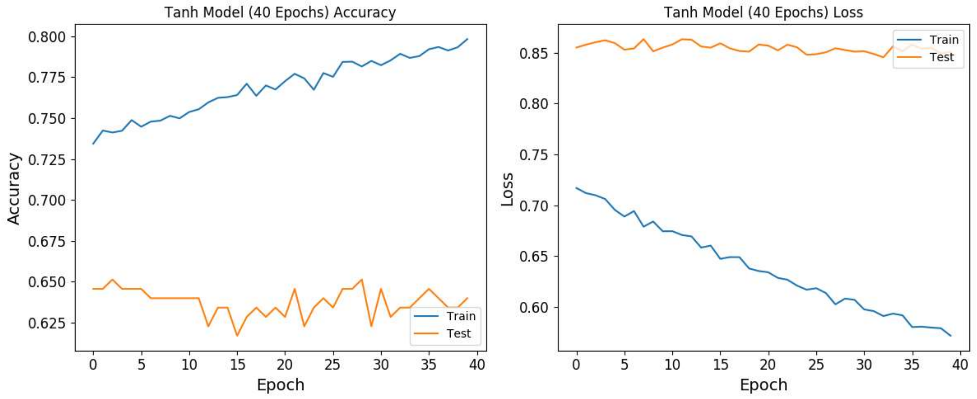
Now we checked for the effect the number of epochs has on our NN. We used the NN with the tanh activation function and checked for 25 epochs vs. 40 epochs. Here are our results:

The accuracy and loss over the testing set for the tanh model with 25 epochs is:



Result of evaluation → Loss = 0.8646, Accuracy = 0.6286

The accuracy and loss over the testing set for the tanh model with 40 epochs is:



Result of evaluation → Loss = 0.8509, Accuracy = 0.6514

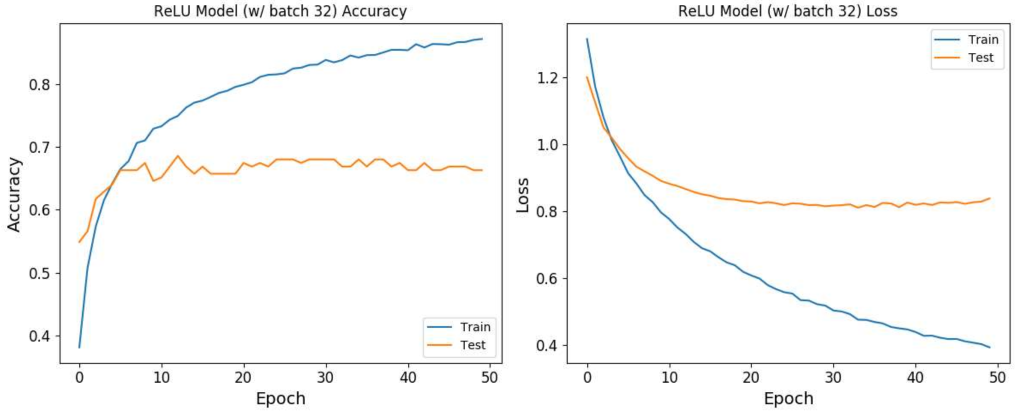
We can see that both the accuracy and loss over the testing set had better results for 40 epochs, but not by a lot ($\Delta loss = -0.0137$, $\Delta accuracy = 2.28\%$). The small change in results (and the generally low accuracy) probably has to do with the low resolution of our sample images, leading to a low ceiling for performance.

Task 4: Mini-batches

The batch size defines the number of training samples fed into the network. Batch gradient descent is where batch = # of samples. Mini-batch gradient descent is where batch < # of samples. SGD (stochastic gradient descent) is where batch = 1. The advantages of SGD over mini-batch gradient descent is that SGD has a better computational speed and uses less memory than mini-batch. The advantages of mini-batch is that it is more accurate. Therefore, we can see that there is a "sweet spot" where we pick the "best" batch size for our data that can be fast computationally, but also provide us with accurate enough results.

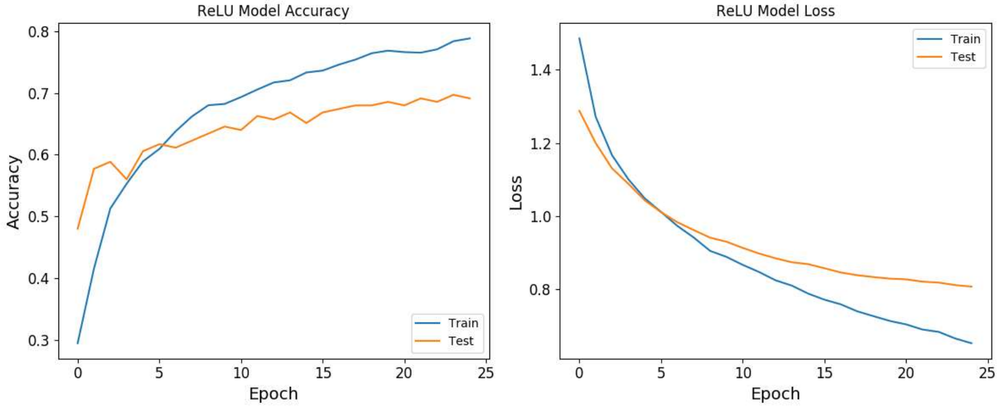
Now we checked the performance of the model after reducing the batch size to 32 (from 64 in the first model we built – task 1). Here are our results:

The accuracy and loss over the testing set for the ReLU model with batch size = 32 is:



Result of evaluation → Loss = 0.8377, Accuracy = 0.6629

The accuracy and loss over the testing set for the ReLU model with batch size = 64 is:



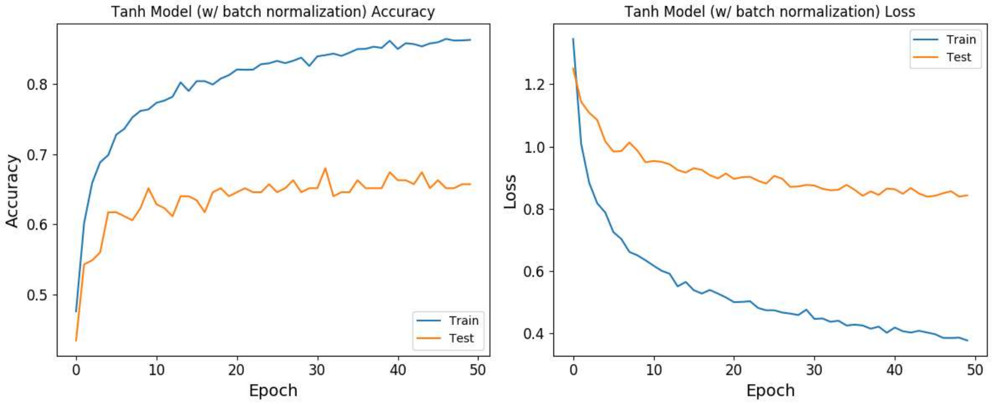
Result of evaluation → Loss =0.8074, Accuracy = 0.6914

We can see a small improvement in the loss and a small decrease in the accuracy. All in all, the results are pretty similar, probably due to the nature of the samples (low resolution images).

Task 5: Batch normalization

Batch normalization is a method where before entering into the next layer, the inputs are normalized and re-centered. This can increase the effectiveness of the NN.

Now, we checked the performance of the tanh activation function model with batch normalization (and 50 epochs). Here are our results:



Result of evaluation → Loss = 0.8429, Accuracy = 0.6571

We can see that there was not a massive improvement when batch normalization was added (and epochs made to be 50). This leads us to believe that due to the nature of the samples (low resolution images) there is a low ceiling for accuracy and loss (around 0.65-0.70 for accuracy, and 0.8-0.9 for loss).

Part 2: Convolutional Neural Network (CNN)

Task 1: 2D CNN

- How many layers does it have?
  - There are 5 2D convolutional layers, 3 pooling layers, 5 batch normalization layers, 6 dropout layers, 1 permutation layer, 1 flatten layer, and 3 fully connected layers. Overall, 24 layers.
- How many filters in each layer?
  - The filters are in the 2D convolution layers.
    - Conv2D_1: 64 filters
    - Conv2D_2: 128 filters
    - Conv2D_3: 128 filters
    - Conv2D_4: 256 filters
    - Conv2D_5: 256 filters
- Would the number of parameters be similar to a fully connected NN?
  - No. We know that for a CNN the number of parameters should be lower (as the filter size increases, the number of parameters decreases), but in out case the CNN has a fully connected NN tail. The NN tail receives a (1,4096) vector, which means that it has 4096*512 parameters (about 2 million). Therefore, this final tail increases the number of parameters in the CNN massively, and creates the vast difference of parameters between the CNN ($p \propto 10^6$) and the previous NN ($p \propto 10^5$).
- Is this specific NN performing regularization?
  - Yes. In every 2D convolution layer there is kernel regularization (L2). Its purpose is to regularize the kernel's weight matrix (using the L2 norm in our case).

Task 2: Number of filters

After changing the number of filters (halving them in each Conv2D layer), we compared the results:

For the CNN with filters = [64, 128, 128, 256, 256] → [test loss, test acc] = [8.2940, 0.3314]

For the CNN with filters = [32, 64, 64, 128, 128] → [test loss, test acc] = [4.856, 0.3028]

When halving the filters, we can see a decrease in loss (although the result is still pretty bad), and in accuracy (about 3%). Therefore, we can see that halving the filters was a bad decision in terms of accuracy and loss.