BME 336546 - HW4

X-ray images classification

Dekel & Nathan

March 1, 2021

1 Fully Connected Layers

1.1 NN with fully connected layers

The accuracy and loss over both the training and testing sets for the first model with ReLu activation are plotted below. The appropriate loss for this multi-class task is categorical cross-entropy.

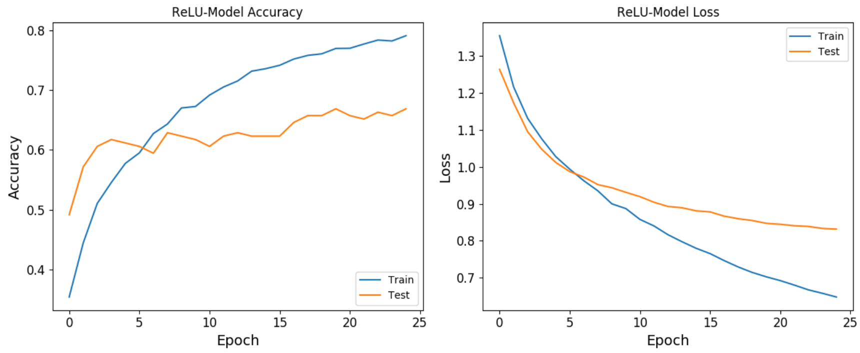

Figure 1: Accuracy and loss against the number of epochs

Results of the evaluation on the test set:

Loss	Accuracy
0.8316	0.6686

Table 1: Accuracy and loss over the test set

1.2 Activation Functions

The activation function determines the output of a neuron given its weighted sum 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:

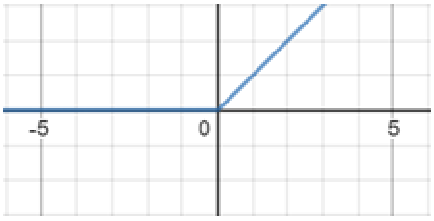

Figure 2: The ReLU function

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 (asymptotically) output as -1 and very positive inputs are output as 1, as shown by the following graph:

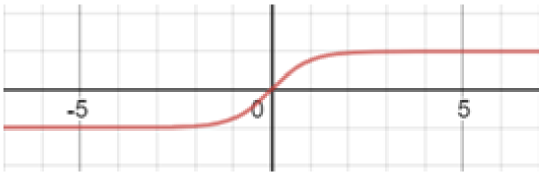

Figure 3: The Tanh function

Tanh is preferred over sigmoid due to its property of centering the data at zero. Nowadays, ReLU is the most extensively used, as both tanh and sigmoid saturate for large inputs, preventing the weights from being updated. Our intuition is that with our normalized data (pixels between 0 and 1), this shortcoming of tanh should not be an important limitation, as its inputs will be of small magnitude, avoiding saturation.

1.3 Number of epochs

An epoch refers to a single pass 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.

25 epochs

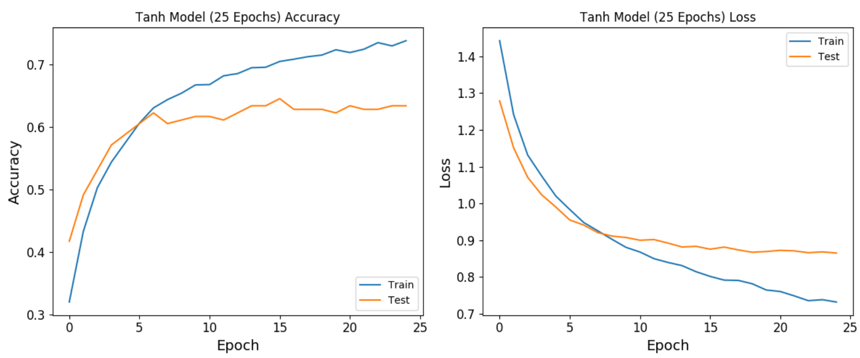

Figure 4: Accuracy and loss against the number of epochs

Results of the evaluation on the test set:

Loss	Accuracy
0.8653	0.6343

Table 2: Accuracy and loss over the test set

40 epochs

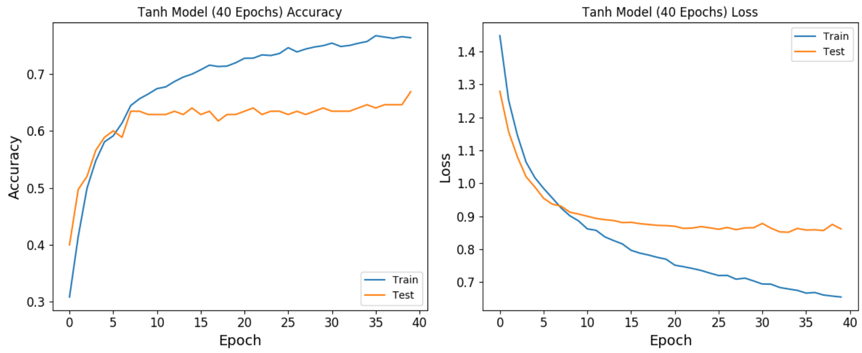

Figure 5: Accuracy and loss against the number of epochs

Results of the evaluation on the test set:

Loss	Accuracy
0.8613	0.6686

Table 3: Accuracy and lost over the test set

First, we don't see saturation effects on the figures above: the algorithm keeps learning. We can notice that both the accuracy and loss over the testing set had better results for 40 epochs, but not by a lot. In Figure 3, we see that the curves for the test set reach a plateau. 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.

Comparing Table 1 (NN with ReLU activation, 25 epochs) and Table 2 (NN with tanh activation, 25 epochs) shows a difference of accuracy of 3.4%.

1.4 Mini-Batches

There are three variants of gradient descent: BGD, SGD, and Mini-Batch Gradient Descent. They differ in how many examples are used to compute the gradient of the cost function.
Let's compare mini-batch (the gradient of the cost function is computed for a subset of the training set) with SGD (the gradient of the cost function is computed for each training example). 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 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:

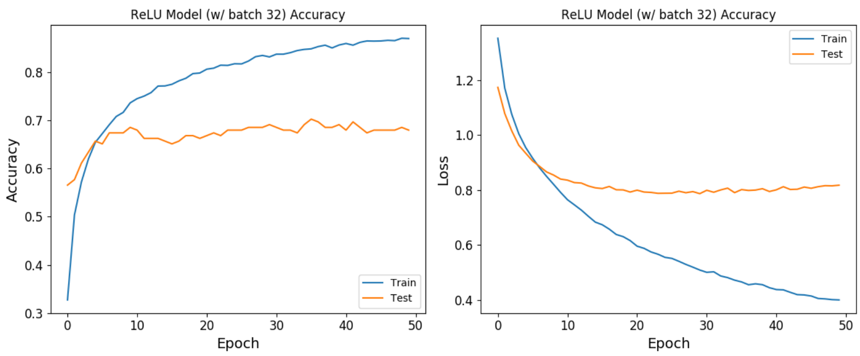

Figure 6: Accuracy and loss against the number of epochs

Results of the evaluation on the test set:

Loss	Accuracy
0.8213	0.6743

Table 4: Accuracy and loss over the test set

1.5 Batch Normalization

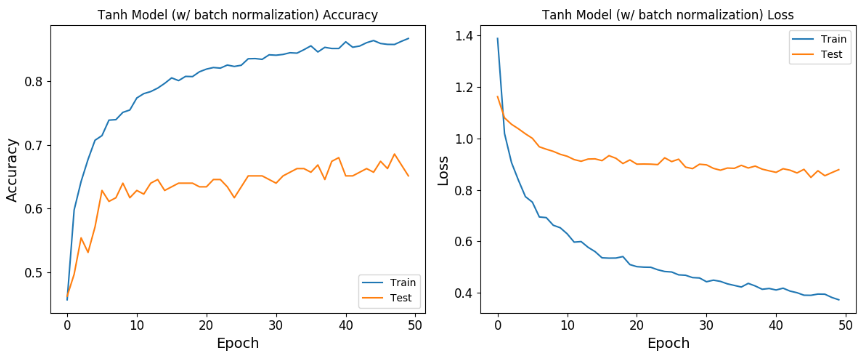

Figure 7: Accuracy and loss against the number of epochs

Results of the evaluation on the test set:

Loss	Accuracy
0.8789	0.6514

Table 5: Accuracy and loss over the test set

2 Convolutional Neural Network (CNN)

2.1 2D CNN

2.2 Number of filters