TIPR Second Assignment
Amal Roy
14989

All the tests were done with training on a fraction (<0.5) of the given dataset and testing on the remaining samples. A minibatch size of 100, learning rate 1e-4 were used and training was done for a maximum of 30 epochs.

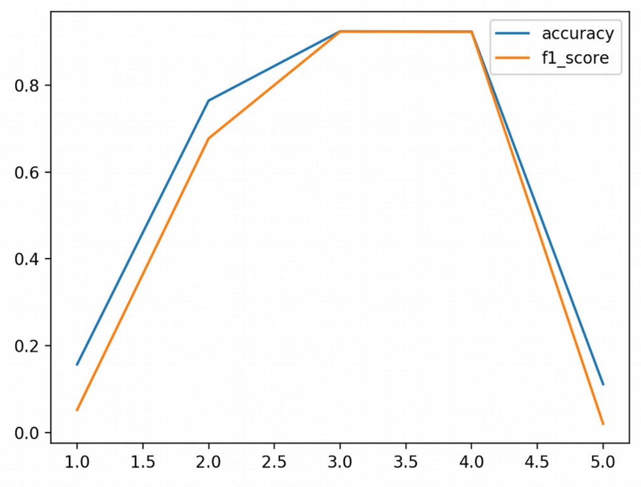
Note for running: the parameter for config is in the format '100,50,20' instead of [100 50 20]

For the Cat-Dog dataset, all images were converted from 200x200 rgb to 28x28 grayscale so that the matrix computations are not very expensive.

A representative plot for each task is given in report. The plots are found in

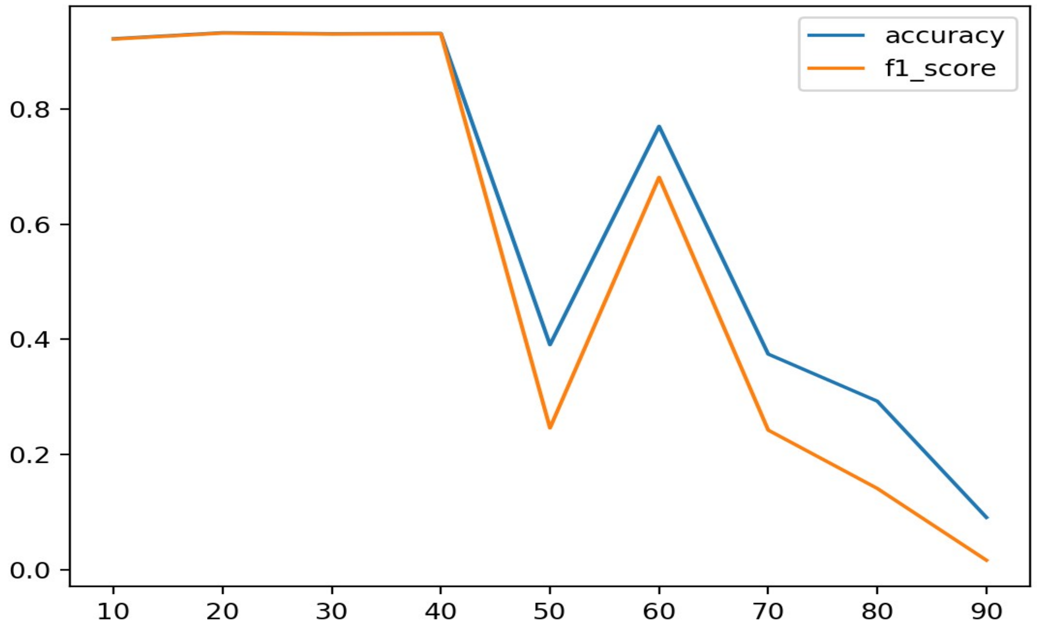
Task 1.

As the plots from MNIST and Cat-Dog datasets, (see fig on right) if the number of hidden layers is one,
the accuracy values are very low. This is possible because we are working with an image dataset and it takes atleast 2 hidden layers to learn features like edges in the image.
If we increase the number of layers too much, the classifier is likely to go towards local minima. In both the datasets, as the number of hidden layers is increased beyond 4, the accuracy reduces considerably.



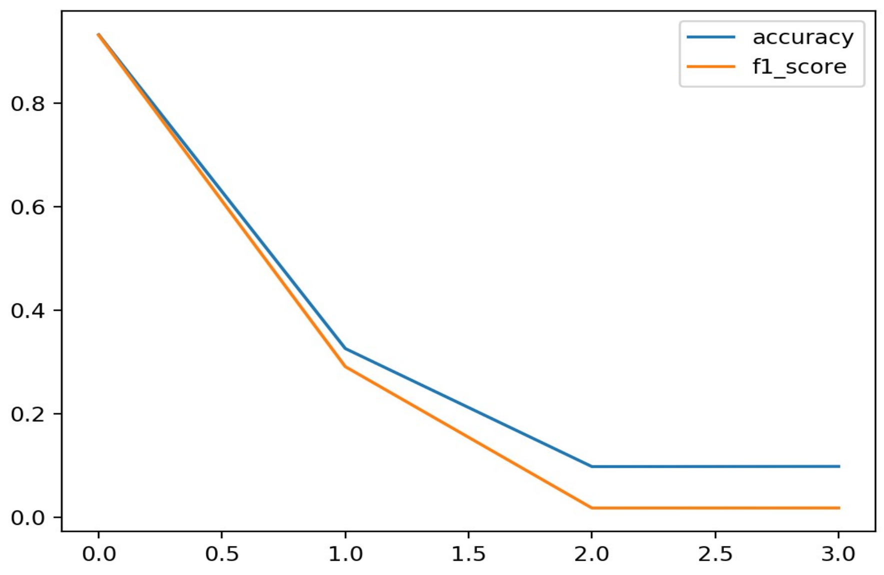
Task 2.

This task was performed in a network with two hidden layers, by fixing the number of neurons in first hidden layer to 100 and changing the number of neurons in second. As the number of neurons is small, the accuracy remains high but as the number of neurons goes near the number of neurons in previous layer, the accuracy takes a hit.



Task 3.

This task was performed on a network of fixed size (3 hidden layers with size 100.50,20) for different activations.



The sigmoid activation gives best result, followed by tanh, swish and relu.
0-sigmoid
1-tanh
2-swish
3-relu
The low accuracy value for swish and relu are obtained because for the normalised input as is used in the implementation, they simply ignore negative values.

Task 4.

If the weights are all initialized to zero, the network cannot learn efficiently. The accuracy value stays the same around the baseline value (0.1 for MNIST and 0.5 for Cat-Dog)
So, the weight matrices were initialized with a Gaussian random varialble with a small variance. This seems to work well in practice.

Task 5.
This task was performed using sequential model in Keras with 2 hidden layers of size 200,50 and my network of the same size. The sequential model is initialized to use Stochastic gradient descent, crossentropy loss and sigmoid activation.

The dataset was split into test and train with 10000 set as number of training samples and the rest of images for testing.
For MNIST, my implementation gave an accuracy of 99.43 for the test set.
The Keras implementation given the same activation function gave an accuracy of 98.65

For Cat-Dog, my implementation gave an accuracy of 75.88
Keras implementation gave 80.45
The low accuracy may be due to downsampling of image to 28x28 and grayscal