Yonat Chen
312245202

Noy Parti
312387798

336546 - Machine Learning in Healthcare

HW4

PART 1: Fully connected layers

Task 2

The activation functions determine the output of the model, its accuracy and its efficiency:
ReLU- is defined as:

$$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

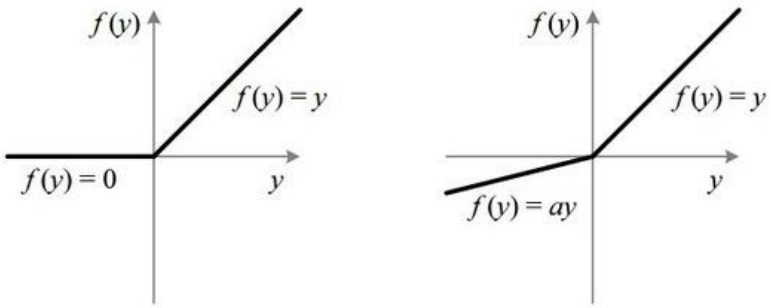

Figure 1:in the right: LeakyReLU , in the left: ReLU

ReLU has output 0 if the input is less than 0, and raw output otherwise.
However, Leaky ReLU modifies the function to allow small negative values when the input is less than zero.
The main disadvantage of ReLU is the negative part that becomes zero and does not take place in the learning ('Dead neurons problem'). Using LeakyReLU solves this problem and allows the learning even with negative inputs. Those two activation functions possess trade-off between computation time and overfitting.
ReLU computation time is often shorter, but its activation function risks with overfitting due to lousy initialization (due to the dead neurons). However, in LeakyReLU the negative part is not zero, and we should not worry about random initialization.

Task 3

When comparing between a model with 25 epochs to a model with 40 epochs we would expect to get a better performance for the model with 40 epochs because than our model has "seen" 15 times more our training set and continued the training. However, we got similar result for the accuracy and loss for both cases.
We think this is because the model has reached its best performance (converged to a local minimum) after 25 epochs and that is why using 40 epochs will not improve our results.

Yonat Chen
312245202

Noy Parti
312387798

Task 4- Mini Batch

In SGD one training sample is passed through the neural network at a time and the parameters (weights) of each layer are updated with the computed gradient.
Due to frequent updates the steps taken towards the minima are very noisy. This can often lead the gradient descent into other directions (local minimum instead of global minimum), it may take longer to achieve convergence, deals with only a single example at a time and therefore the computations are longer.
However, Mini-batch gradient descent is a variation of the gradient descent algorithm that splits the training dataset into small batches that are used to train the data.
it is computationally efficient, less noisy, has a faster convergence and more accurate (converge to a global minimum).

Task 4- Batch normalization

Batch normalization is used to get faster and more stable layers by normalization.
it should improve the performance, however in our case the performance stays similar and even decreases.

PART 2: Convolutional Neural Network (CNN)

Task 1:

- There are 8 layers.
- the first 5 layers contains filters:
    - first layer: 64 filters
    - second layer: 128 filters
    - third layer: 128 filters
    - fourth layer: 256 filters
    - fifth layer: 256 filters

  layers 6-8 does not include filters.

- The number of parameters will be lower than fully connected NN because the Conv2D CNN model uses max pooling (part of the filters). that way the number of dimensions is reduce.
- This specific NN performs regularization because every layer includes l2 regularization. Moreover, if the input argument "drop" is True the model will perform "Droupout" in order to have a regulation.

Task 2:
The results in both cases are not very good- the accuracy is low (22-36%), that can happen due to bad data so that the filters number will not have a big influence or a neural network that is not suitable for this classification.

for the new filters (cut by half) the results have been improved. We believe this is because when we used the original filters their sizes caused over-fitting of the model, and that is why the loss and accuracy have improved in the reduced filters (which are not over-fitted).