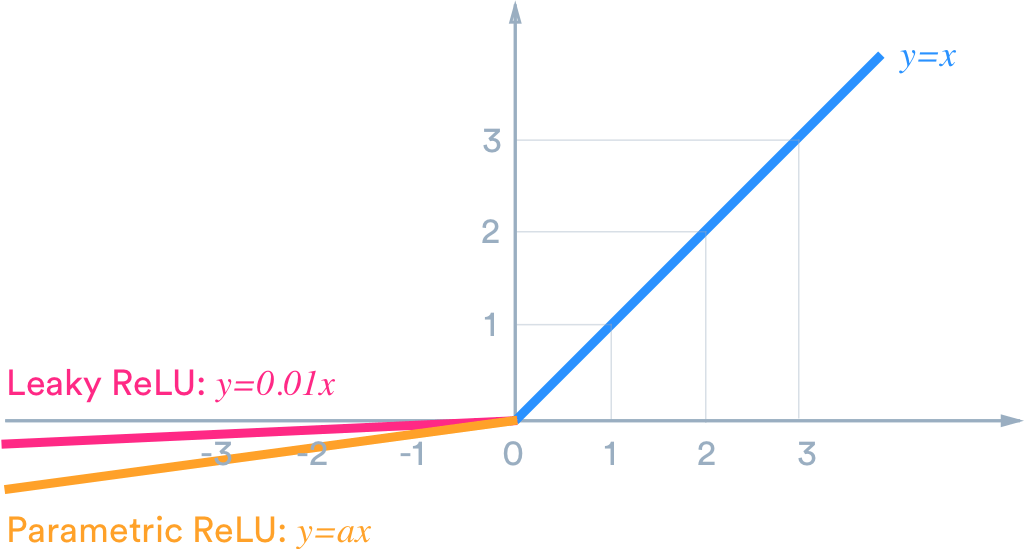
***Task 2:*** *Activation functions.*

Change the activation functions to LeakyRelu or tanh or sigmoid. Name the new model new\_a\_model. Explain how it can affect the model.\*

We change the activation function to Leaky- ReLU:



This change will affect the model non-linearity method.

The Leaky ReLU has some benefits over the ReLU[1]:

1. It does not have a zero-slope parts.
2. It speeds up the training because unlike ReLU, leaky ReLU is more “balanced,” and may therefore learn faster.

***Task 3:*** *Number of epochs.*

Train the new model using 25 and 40 epochs. What difference does it makes in term of performance? Remember to save the compiled model for having initialized weights for every run as we did in tutorial 12. Evaluate each trained model on the test set

When the number of epochs is more than the necessary, the model will learn patterns that are specific to the training dataset and we may get **overfitting**. Thus, the model incapable to perform well on new set of data (the test dataset in our case) – the accuracy on the training data will be high but for **the testing data the accuracy will decrease[2]**.

On the other hand, too small number of epochs will cause underfitting, the model will not learn enough feature before the train end and our results will be low for both the training ang testing data.

As a result, the model should be trained for an optimal number of epochs.  
from our training, we can notice that the model which was trained with 25 epochs achieve better accuracy score than the model with 40 epochs.

***Task 4:*** *Mini-batches.*

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?\*

***Task 4:*** *Batch normalization.*

Build the new\_a\_model again and add batch normalization layers. How does it impact your results?\*

Part 2 :

***Task 1:*** *2D CNN.*

Have a look at the model below and answer the following:

* How many layers does it have?
* How many filter in each layer?
* Would the number of parmaters be similar to a fully connected NN?
* Is this specific NN performing regularization?

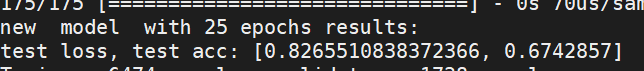
***Task 2:*** *Number of filters*

Rebuild the function get\_net to have as an input argument a list of number of filters in each layers, i.e. for the CNN defined above the input should have been [64, 128, 128, 256, 256]. Now train the model with the number of filters reduced by half. What were the results.

Tests results :

|  |  |  |
| --- | --- | --- |
| acc | loss | model |
| 0.662 | 0.811 | ReLU model, batch = 64 |
| 0.674 | 0.826 | New model, 25 epochs |
| 0.645 | 0.834 | New model, 40 epochs |
| 0.64 | 0.838 | ReLU model, batch = 32 |
| 0.634 | 1.061 | New model, batch norm |
| 0.377 | 7.788 | NNet model |
| 0.291 | 4.652 | NNet reduced filters |

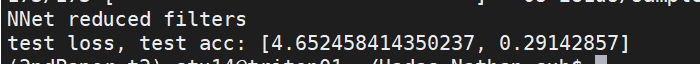












References:

[1] <https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7>

[2] <https://www.geeksforgeeks.org/choose-optimal-number-of-epochs-to-train-a-neural-network-in-keras/>