**Assignment 2**

Group members: Una Sekulović – 12303562,

1. At the beginning we got familiar with our dataset. After we loaded data, we used the *ydata\_profiling* library in order to get a better insight into our data, ie we looked at how our data is distributed - minimum, maximum, mean etc., and the library also visualizes each feature. We split the train set into a train (80%) and a validation set (20%) using the *train\_test\_split* function from the *sklearn.model\_selection* library. Finally, we normalized our input data using the *preprocessing.* *StandardScaler* function. This function normalizes each input feature to have zero-mean and unit-variance.
2. We decided that the output layer has one node because it is a regression task and the goal is to predict one target variable (values of the houses) and it doesn't have an activation function because we want the output to take an arbitrary value. For the loss function, we decided to use the Mean Squared Error, because it is the loss function that is most used for the regression task and it also has the advantage of penalizing large errors and is therefore more appropriate in different situations. Below is a table in which the training loss and validation loss for different architectures (varying numbers of hidden units and hidden layers) are presented:

|  |  |  |
| --- | --- | --- |
|  | Training loss | Validation loss |
| 5 hidden layers, every layer has 64 units, activation is relu, epoch=10 | 0.3368 | 0.3799 |
| 4 hidden layers, every layer has 64 units, activation is relu, epoch=10 | 0.3419 | 0.3084 |
| 5 hidden layers, first and second layer have 64 units, third and fourth layer have 24, fifth has 12 units, activation is relu, epoch=10 | 0.3544 | 0.3432 |
| 5 hidden layers, first, second and third layer have 64 units, fourth and fifth layer have 24, activation is relu, epoch=10 | 0.3564 | 0.3148 |
| 5 hidden layers, every layer has 64 units, activation is softmax, epoch=10 | 1.3502 | 1.2945 |
| 5 hidden layers, every layer has 64 units, activation is sigmoid, epoch=10 | 0.3923 | 0.3656 |
| 5 hidden layers, every layer has 64 units, activation is sigmoid, epoch=100 | 0.2294 | 0.2805 |

After testing different examples we decided that our neural network has 5 hidden layers and that each layer has 64 neurons and the activation function we use is ReLU.



|  |  |  |
| --- | --- | --- |
|  | Training loss | Validation loss |
| Optimizer = Adam, epoch=10 | 0.3514 | 0.3231 |
| Optimizer = SGD, epoch=10 | 0.3803 | 0.3747 |
| Optimizer = SGD with momentum = 0.9, epoch=10 | 0.3635 | 0.3807 |
| Optimizer = Adam, epoch=100 | 0.2029 | 0.2551 |
| Optimizer = SGD, epoch=100 | 0.2695 | 0.2858 |
| Optimizer = SGD, momentum= 0.9, epoch=100 | 0.2521 | 0.2669 |
| Optimizer = Adam, learning rate=0.01, epoch=100 | 0.2804 | 0.2749 |
| Optimizer = Adam, learning rate=0.0001, epoch=100 | 0.3129 | 0.3199 |
| Optimizer = Adam, learning rate=0.1, epoch=100 | 1.3604 | 1.2941 |
| Optimizer = Adam, learning rate=0.001, epoch=100 | 0.1952 | 0.2858 |
| Optimizer = Adam, learning rate=1, epoch=100 | 1.4039 | 1.3006 |
| Optimizer = Adam, learning rate=0.005, epoch=100 | 0.2375 | 0.2557 |
| Optimizer = Adam, learning rate=0.003, epoch=100 | 0.2127 | 0.2455 |
| Optimizer = Adam, learning rate=0.0025, epoch=100 | 0.2092 | 0.2537 |
| Optimizer = Adam, learning rate=0.0035, epoch=100 | 0.2474 | 0.2703 |
| Optimizer = Adam, learning rate=0.00275, epoch=100 | 0.2533 | 0.2673 |