1 Multi-Layer Perceptron - Part 1

1.1 Learning gate investigation - a

In the following point it was performed investigation of the learning rate for stochastic gradient descent. Tests was performed on the different range of values of the learning rate that differs by the power of 10 (0.00001, 0.0001, 0.001, 0.01, 0,1) with the stable and decreasing value of the learning rate. This method of matching the parameter is very effective, because it is easy to note the differences that occurs in the model. In the Fig.1 it can be seen the result of training.

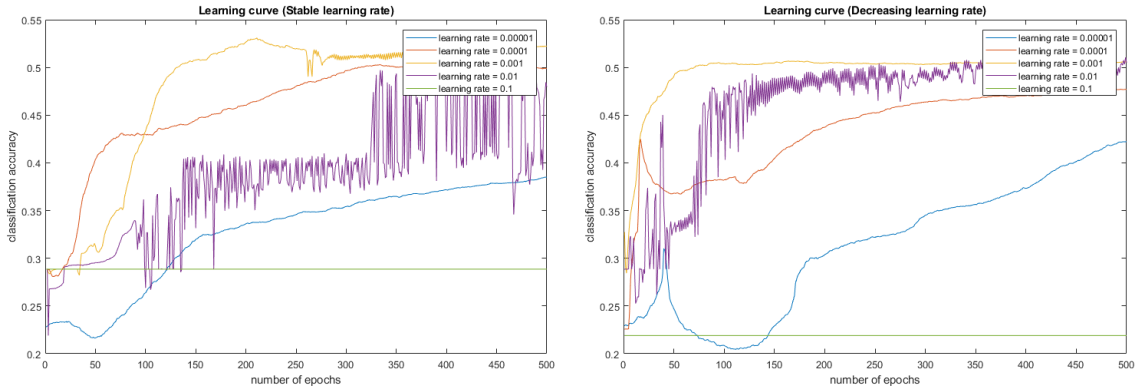

Figure 1: Plots of the model with the different learning rate

It can be easily noticed that dynamic learning rate (gradually decreasing - plot on the right) help in faster training, while the stable one is much slower when it comes to reaching the best accuracy. In the Fig.1 on the first plot (one with Stable Learning Rate = on the left) it can be noticed some disturbances in the learning curve which are caused by to big value of the learning rate which overshoot the minimum instead of stabilize at the end. Stable learning rate makes constant steps toward the minimum, but the problem occurs when it tries to reach minimum. To big steps make it impossible that is why disturbances occurs. Situation is slightly different on the right plot where disturbance also occurs (for the learning rate = 0.01) however during increasing number of epochs the learning curve become more and more stable due to constantly decreasing value of the learning rate. When the learning rate is too small it can be a slow convergence problem. It means that there is needed huge amount of epochs to get to the local minimum which far from the global one. With to small learning rate there is also problem that can stuck in the local minimum. When the learning rate is too large the cost function may not decrease on every iteration and may not even converge. In some cases when the learning rate is too large the slow convergence also occurs.

Summing up stable learning rate is fast when it comes to going toward local minimum but there is huge chance of the overshoot, while dynamic one is a little bit slower but there is great chance that will get to the local minimum. Small learning rate might never cause the result that local minimum would be never reached, while to large learning rate might cause overshoot.

1.2 Network geometry investigation - b

Next task investigate different configurations of the Hidden layer. For this problem it has been performed calculations for configuration with the one layer (containing from 1 to 10 neurons) and two layers (every possibility from 1 to 5 neurons). The reason of such a choice is that science papers gives information that 90% of cases and problems where the neural networks are used can be solved using one or two layers. Multilayer perceptron is time consuming method, so for the bigger amount of data is important to choose wise the configuration. For the case with more data investigation of the best possible configuration of the hidden layers would not be possible due to huge amount of time needed for the computation. What is also important is that more layers means greater chance to overfit the model. In the Fig.2 it can be seen the result.

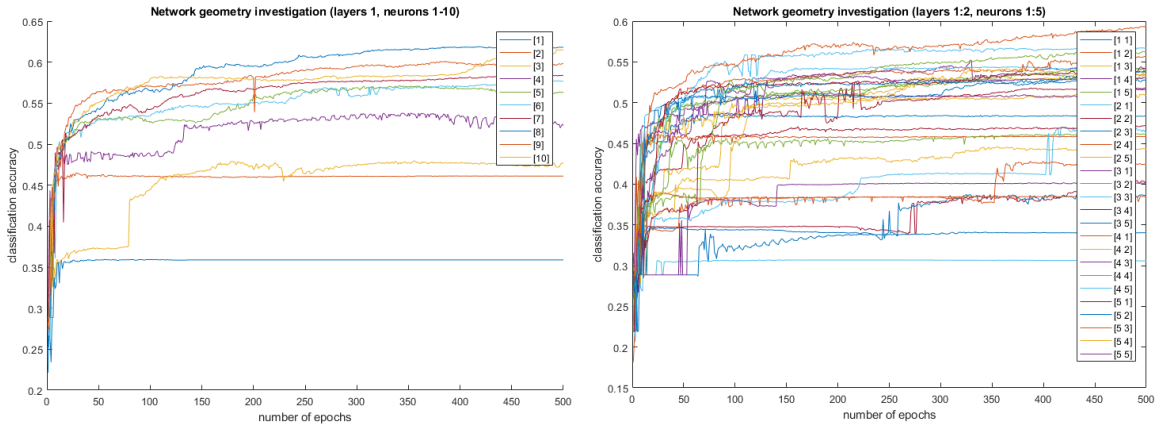

Figure 2: Plots of the model with the different Hidden layer configuration

It can be noticed that number of layers have influence the accuracy of the model. Curves of the one-layered model are more smooth. Too few neurons in the hidden layers results in disturbances which are produced by underfitting. While too many neurons in the hidden layers can result in overfitting, which happen when the neural network has so much information processing capacity that the limited amount of information contained in the training set is not enough to train all of the neurons in the hidden layers. Too many neurons can also

increase training time to the point that it is impossible to adequately train the neural network. Generally greater amount of neurons improve accuracy because every weight is able to generate and optimise better abstractions from the input data while.

When it comes to number of layers it is important to notice that bigger amount of layers gives big values of the parameters which means that more time is required for training. With large amount of hidden layers there is also chance to overfit the model. That is why it is better to use smaller amount of layers.

1.3 Choosen MLP configuration - c

The best learning rate for this model is gradually decreasing one with the value equal 0.001, because the curve obtain the best accuracy and the the time needed for training is very small (about 70 epochs). In the Fig.3 it is possible to see the table with the maximum accuracy and computation time obtained for each configuration of the hidden layers in the model.

Hidden Layer	1	2	3	4	5	6	7	8	9	10	[1,1]	[1,2]	[1,3]	[1,4]	[1,5]	[2,1]	[2,2]	[2,3]
Max Axxuracy	0.3594	0.4648	0.4796	0.5378	0.5712	0.5778	0.5844	0.6188	0.6012	0.6154	0.3068	0.4052	0.3874	0.3848	0.4458	0.4016	0.4618	0.47
Time	111.603	131.7474	156.1784	141.8623	151.629	213.1694	250.5048	252.7163	253.8042	263.3227	145.98	152.3889	157.6931	164.3797	169.2298	152.7253	158.182	163.2802

Hidden Layer	[2,4]	[2,5]	[3,1]	[3,2]	[3,3]	[3,4]	[3,5]	[4,1]	[4,2]	[4,3]	[4,4]	[4,5]	[5,1]	[5,2]	[5,3]	[5,4]	[5,5]
Max Axxuracy	0.4718	0.4866	0.428	0.5316	0.5196	0.5362	0.544	0.5258	0.529	0.5574	0.542	0.536	0.5422	0.5676	0.544	0.539	0.5938
Time	169.4197	175.3995	163.1136	192.1348	323.425	232.4945	284.414	182.7189	217.4695	230.0891	216.0214	245.2113	169.4641	176.8944	225.5151	191.6622	193.4509

Figure 3: Table with the different configurations of the Hidden Layers, maximum accuracy and duration

When it comes to configuration of the hidden layers the highest accuracy is when the model has 1 layer and 8 neurons. Curve of that configuration is stable and easily obtain maximum accuracy. One layer is also safe choice, because is small chance to overfit the model.

Chosen configuration is: learning rate: decreasing 0.001, hidden layers: one layer with 8 neurons.
To obtain the confusion matrix function MLP_REST was used (this function do not get learning rate as the input, that is why only configuration of the hidden layer was adjusted.). In the Fig.4 is presented the result (Confusion matrix with frequencies)

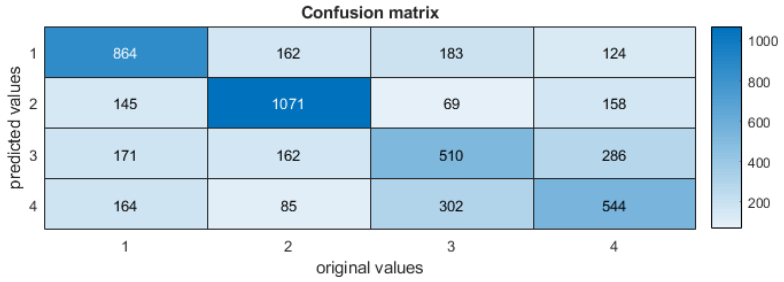

Figure 4: Confusion matrix for the chosen configuration of the model

Selected configuration of the neural network has 56 weights (Such a number was obtained by the sum of multiplication of the input data (3) and number of neurons in the hidden layer (8) with multiplication of the number of neurons of the hidden layers (8) and the number of labels (4))

$$w = 3 * 8 + 8 * 4 = 56$$

2 Multi-Layer Perceptron - Part 2

For this task two classes have been chosen: first and second one. The second performs the best because is the most dispersed, due to that it is simple to classify because there is no data from other classes that could disturb the results. While the first class performs also very good because is very condensed. Result are presented in the Fig. 5.

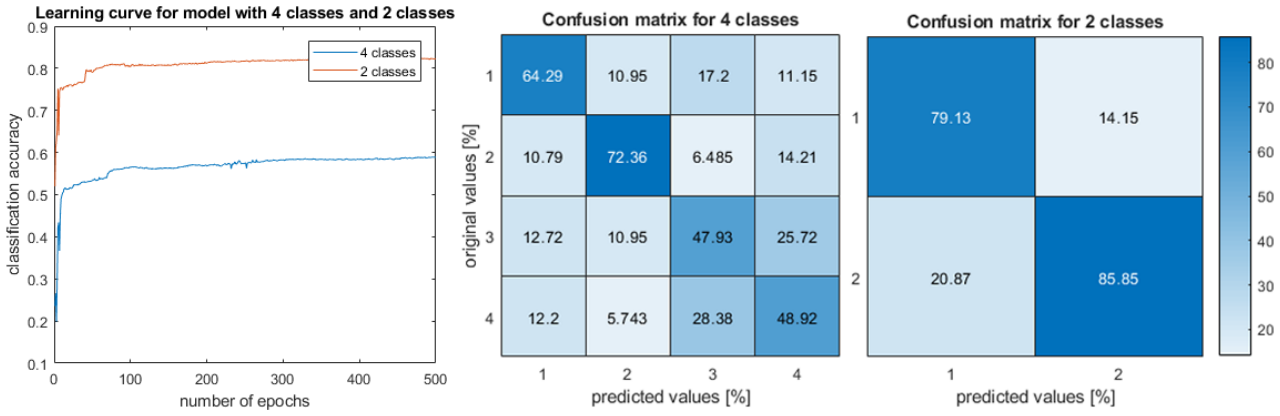

Figure 5: binary classification of the class 1 and 2 compared to the previous classifcation

For the better visualisation confusion matrix values were change from frequencies to probabilities (That was obtained by dividing each number by the corresponding sum of the predicted values, in this case sum of column). Thank to that it can be noticed that classification with 2-classes performs better than 4-classes. Maximum accuracy of the new one is equal 82.35%.

3 Classifier - Part 3

First method was used to compare Multilayer perceptron was Probabilistic model classification (Matlab code in the appendix). In this method have been used Naive Bayes theorem. The idea of this theorem is that there is calculated probability of the label one given point x and probability of the label two given point x. On the basis of that two probabilities the higher one classify the label for the point. Such a calculations are performed for each point from the test data to perform classification. This method is parametric method where parameters are mean and sigma of the gaussians. Maximum accuracy obtained from that method was equal 78.78%.

Due to smaller accuracy in the Probabilistic model classification method than in the Multilayer Perceptron method the third method have been performed: K-Nearest neighbour. Methodology of the second method is as follows. The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabeled vector (test data) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. KNN is non-parametric method, because is not based on the model but based on the data. Hyperparameter of this method is K which can be adjusted depending on the cases. In this case maximum accuracy of the KNN method was obtain using K = 5 equal 86.96% (Fig.6).

K parameter	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Max Axxuracy [%]	85.84	85.84	86.96	86.64	86.53	86.28	86.49	86.10	85.95	85.99	85.99	86.10	85.48	85.37	85.37	85.12	85.16	84.91

Figure 6: Maximum accuracy using KNN method for the different value of the hyperparameter K

4 Classifier - Part 4

Comparison of the two methods is presented in the Fig. 7.

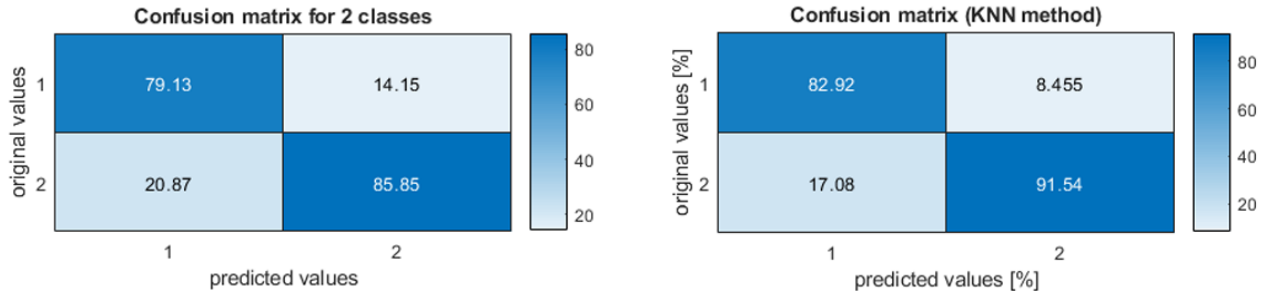

Figure 7: Confusion Matrix for the MLP and KNN method for 2 classes

Looking at two confusion matrices it can be noticed that KNN method classify better both the first and the second class. Summing up in the Fig. 8 are presented advantages and disadvantages of the three classification methods used in this report.

Multi Layer Perceptron	Probabilistic model classification	K-Nearest neighbour
Can be used for difficult to complex problems Good algorithm to use for the regression and mapping	Fast to train Can handle missing data very well	Fast to train Easy to implement Easy to tune (only K) Easy when characteristics are unknown
Algorithm can get stuck in a local minima Difficult to train Very slow to train	Can be fooled by zero frequencies Can't learn interactions between features	Slow during prediction (not training) Doesn't handle missing data gracefully Doesn't know which attributes are more important
Maximum obtained accuracy = 82.35%	Maximum obtained accuracy = 78.78%	Maximum obtained accuracy = 86.96%

Figure 8: Table with comparison of the three methods

There is no right answer which method is the best. It important to chose wise taking into the consideration first of all the type and amount of data that have been given, type and complexity of data that need to be classify, power and computation possibilities of the computer that will be used for the calculations.