Polytechnic University of Catalonia

Data Science and Engineering

Pattern Recognition with Single Layer Neural Network

Alina Castell Blasco - 49188689S

Abril 2023

Contents

1 Introduction **2**

2 Study of Convergence **2**

2.1 Global Convergence 2

2.2 Local Convergence 5

2.3 Discussion of Convergence Study 10

3 Study of recognition accuracy **11**

3.1 Best Method analysis 12

3.2 Discussion of $Accuracy^{TE} - \tilde{L}$ with respect to λ-algorithm 14

1 Introduction

This project describes the development of an application of recognition of numbers in a sequence of blurred digits. The procedure to achieve this goal is to apply Single Layer Neural Network (SLNN) that is trained to identify this numbers with First Derivative methods, all based on the unconstrained optimization algorithms studied in the course. These methods are the Gradient (GM), the Quasi-Newton (QNM) together with the BFGS and the Stochastic Gradient (SGM).

A code 'uo nn solve' has been generated in order to recognize numbers with these methods; with the information provided and comparing all three methods, on of them has been determined to be the most adequate to complete the task of the project. This decision has been made with the help of the global and local convergence and the accuracy rate. Note that the data set has been generated with the seeds trseed = 491886; teseed = 234789; sgseed = 565544;.

Let us start the study of a recognition of numbers algorithm.

2 Study of Convergence

In this section we are going to study the global and local convergence of GM, QNM and SGM in terms of the objective function (average loss function). By analysing these results we can obtain the best regularization parameter λ and the best algorithm in terms of global and local convergence.

2.1 Global Convergence

The global convergence states the capacity of convergence of any combination of methods. Accordingly, we need to check whether a method reaches the optimal solution; in this case, when the objective function recognises the target number.

We define that a method converges globally if the optimal solution is obtained before 1000 iterations, in the GM and QNM methods. In the SGM case, the condition is just if an optimal solution is obtained. Therefore, next, we show three tables with the number of iterations per λ-value and method. Is there any case where there is no convergence?

$la = 0$	**GM**	**QNM**	**SGM**
1	68	35	125126
2	160	35	1501
3	1001	62	1626
4	43	35	125126
5	112	35	4251
6	585	35	2251
7	76	35	12251
8	1001	35	1626
9	1001	35	1876
0	502	550	1876

Table 1: Number of iterations for lambda = 0.

When lambda is 0:

- Gradient Method there are three targets (*numtarget=3, numtarget=8, numtarget=9*) where the algorithm is not capable of identifying a stationary solution of the loss function.
- Quasi-Newton Method all optimal solutions are reached with a low number of iterations.
- Stochastic Gradient Method, it has convergence for all target numbers.

$la = 0.01$	**GM**	**QNM**	**SGM**
1	55	41	1876
2	118	44	1501
3	198	52	3876
4	51	43	2876
5	93	41	2126
6	137	46	1376
7	57	47	1876
8	335	49	1751
9	204	44	2501
0	154	51	1626

Table 2: Number of iterations for lambda = 0.01.

When lambda is 0.01:

- Gradient Method is globally convergent because minimums are reached before the 1000 iterations for all targets.
- Quasi-Newton Method is as convergent as GM.
- Stochastic Gradient Method is convergent for all targets as well.

Also, we can observe that the number of iterations is smaller with respect to the previous lambda value.

$la = 0.1$	**GM**	**QNM**	**SGM**
1	35	35	1376
2	47	35	1626
3	63	35	2376
4	35	35	3376
5	49	356	2626
6	58	35	3626
7	35	35	2626
8	74	35	2751
9	70	35	1376
0	59	35	1501

Table 3: Number of iterations for lambda = 0.1.

When lambda is 0.1 the results are very similar to the previous results with λ=0.01. All methods reach global convergence and with it the optimal solution. In addition, notice that the number of iterations is considerably constant for each method and greatly reduced with λ=0.1 for methods GM and QNM, while for SGM is similar to λ=0.01.

To summarize, we can affirm that the best method in terms of less number of iterations and more number of targets where there is global convergence is the QNM.

In order to better visualize the best combination of method and regularization parameter, we show a table of the average of the values obtained by the loss function ($\tilde{L}$):

lambda	**GM**	**QNM**	**SGM**
0	0.0122	0.3480	0.3552
0.01	0.063	0.3491	0.4352
0.1	0.1950	0.3206	0.6229

Table 4: Average of the obtained results from the loss function L*

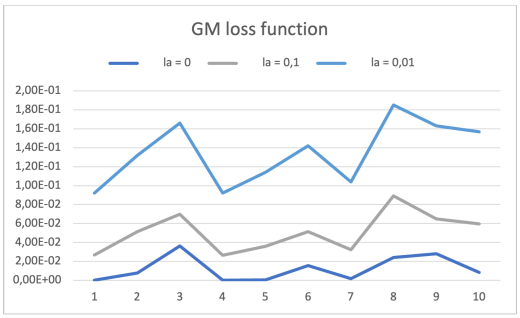
The table shows the average values of the objective function that each of these methods obtains, once the optimal solution is reached. At first, what we observe is that the GM is the one that obtains values closer to 0; the QNM has stability within all λs; the SGM obtains values further from the 0 than any other. In general, the higher the value of λ, the higher the average value of the objective function. As for declaring which is the best method in terms of $\tilde{L}$ values, the minimum values are obtained with the Gradient Method, regardless the lambda.

Knowing this, we can declare the best methods for each λ, where 'best method' is the one with lower value of the loss function $\tilde{L}$ in average and that has global convergence.

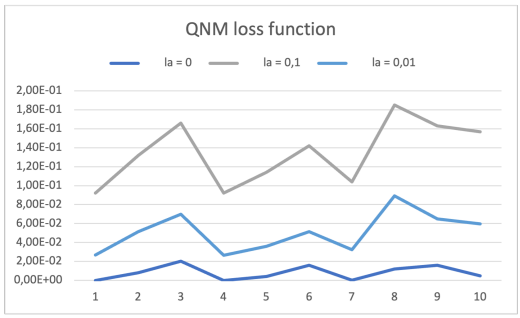
- Case $\lambda = 0$: QNM.

- Case $\lambda = 0.01$: GM.
- Case $\lambda = 0.1$: GM.

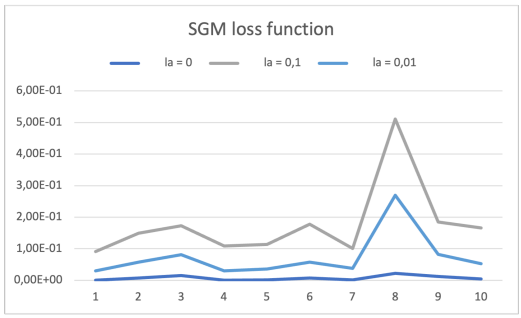
Additionally, we provide the graphical representation with respect to lambdas of the value of the loss function of every target.



(a) GM



(b) QNM



(c) SGM

Figure 1: Values of the loss function in terms of λ for each target and method

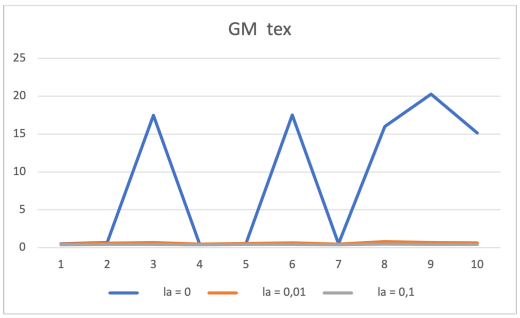
After observing the graphics, we see that for $\lambda = 0$ all algorithms reach different solutions, the loss functions has little variation around the optimum and all methods reach minimum $\tilde{L}$; this is because no regularization is applied. On the contrary, in the cases where $\lambda = 0.01$ and $\lambda = 0.1$ a regularization is applied to the method and so to the $\tilde{L}$ is added the norm of the solution vector (w); there is much more variation within the same method.

To conclude this brief study, we can extract that the global convergence property does not depend on the value of the regularization parameter, while the value of the loss function $\tilde{L}$ does.

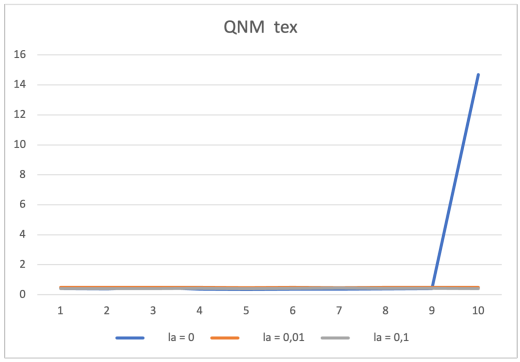
2.2 Local Convergence

In order to study the speed we will take into account the execution time and the number of iterations, we will see if it depends on the lambda value. We will consider that the best local convergence is

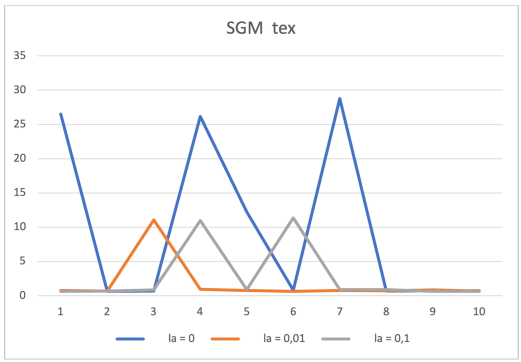
the one with less execution time per iteration. First, we show the graphical representation with respect to lambdas of the execution time of every target.



(a) GM



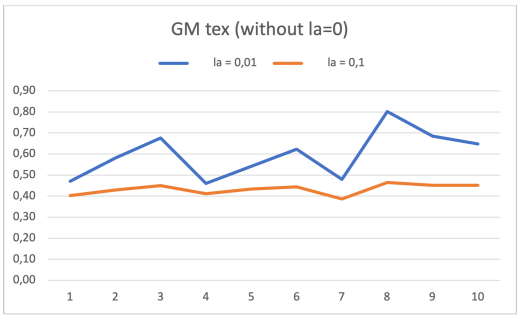
(b) QNM



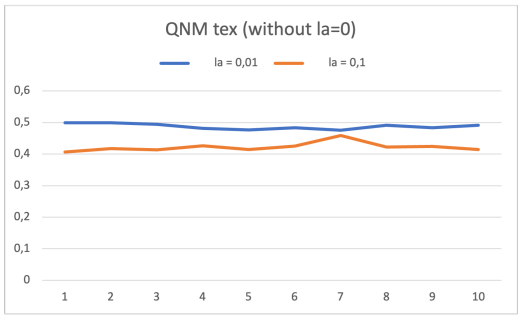
(c) SGM

Figure 2: Execution time of λ for each target and method

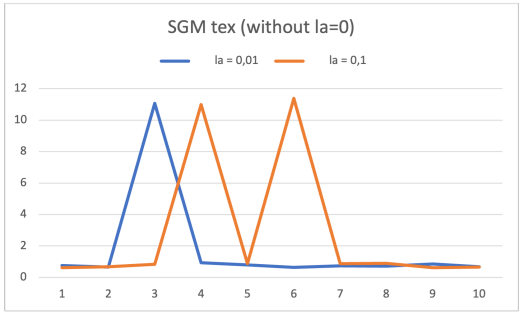
The first two methods and its graphics are not easy to visually analyse because of the high execution times when using a regularization parameter of λ=0. So, here we show the graphics without the λ=0 parameter, so as to be able to see the behaviour of the other lambda values.



(a) GM



(b) QNM

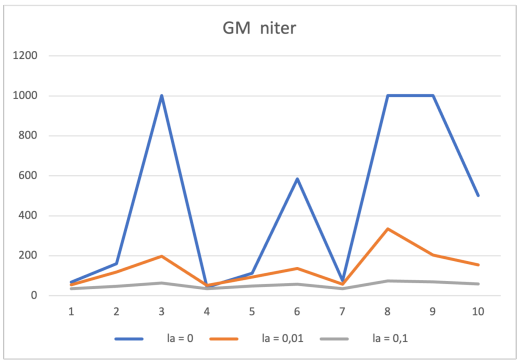


(c) SGM

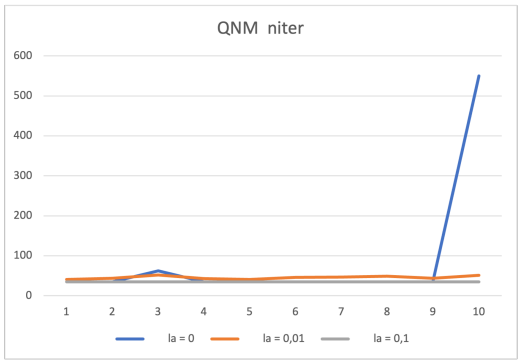
Figure 3: Execution time of λ=0.01 and λ=0.1 for each target and method

We can see that the methods GM and QNM have higher execution times for λ=0.01 in all target numbers (note that there is a change of colours in the lines of this chart and the previous one, the legend shows the value assigned). Of all methods, the SGM is the one with higher execution times generally,something that we could also observe in the previous chart. In general, the higher the λ parameter the smaller the execution time.

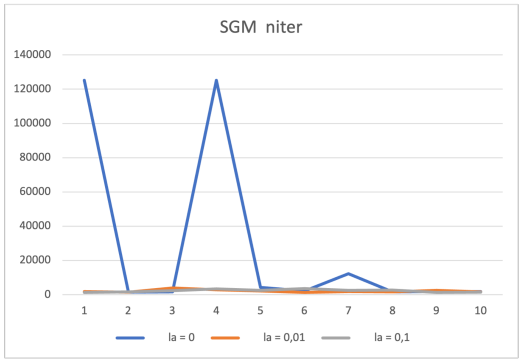
Secondly, we show the graphical representation with respect to lambdas of the number of iterations of every target. In this case, is easier to observe the differences between lambda values within the graphic.



(a) GM



(b) QNM



(c) SGM

Figure 4: Execution time of λ for each target and method

Similarly to the execution time, the λ=0 is the value that generates higher number of iterations for all methods; although, notice that in QNM method the number of iterations is very similar with all target numbers except the number 0 (or 10 as shown in the chart). Again in general, the higher the lambda parameter the lower the number of iterations.

So as to further analyse this convergence, we will study the ratio tex/niter which will determine which method is better. We represent a table with the average number of iterations, the average time of execution and the relation between them of all values of lambda.

	Average niter	**Average tex (s)**	**Average tex/niter (s)**
GM	215,86	3,315	0,0153
QNM	56,666	0,9113	0,016
SGM	10738,5	20,613	0,0019

Table 5: Average number of iterations and execution time for each method.

Observing the ratio tex/niter we might believe that the SGM is the best method since it has the

lowest ratio, this means that it will converge more rapidly in terms of execution time per iteration.

Next, we will study the average ratio tex/niter differentiating the lambda values by presenting similar as the previous one.

Case λ=0:

λ=0	**Average niter**	**Average tex (s)**	**Average tex/niter (s)**
GM	454,9	8,9162	0,0148
QNM	89,2	1,824	0,0122
SGM	27751	57,209	0,0118

Table 6: Average number of iterations and execution time for each method.

At first sight we can say that the number of iterations of all methods is approximately double the number in the previous table of average values. This is due to the fact that the objective function tends to be plane around the minimum value, causing the distance between iterations of the algorithm very small. From this table we can obtain that the method with better ratio tex/niter (smaller) is the SGM, even though, it is the one with greater number of iterations and execution time. Despite that, it is fair to say that all three methods have very similar ratios, so very similar local convergence.

Case λ=0.01:

λ=0.01	**Average niter**	**Average tex (s)**	**Average tex/niter (s)**
GM	140,2	0,5968	0,0055
QNM	45,8	0,4874	0,0107
SGM	2138,5	1,7850	0,0064

Table 7: Average number of iterations and execution time for each method.

In this case, the ratio is very similar for methods GM and SGM and ten times smaller than for the QNM; this time the GM is a bit better than the SGM. Then again, this last method is the one with greater number of iterations and execution time. Then, for this case, GM is the one with better local convergence.

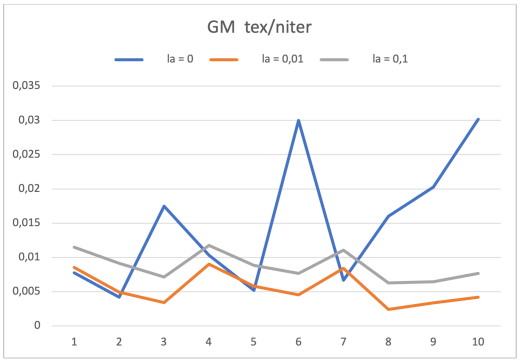
Case λ=0.1:

λ=0.1	**Average niter**	**Average tex (s)**	**Average tex/niter (s)**
GM	52,5	0,4323	0,0087
QNM	35	0,4222	0,0121
SGM	10738,5	20,613	0,0874

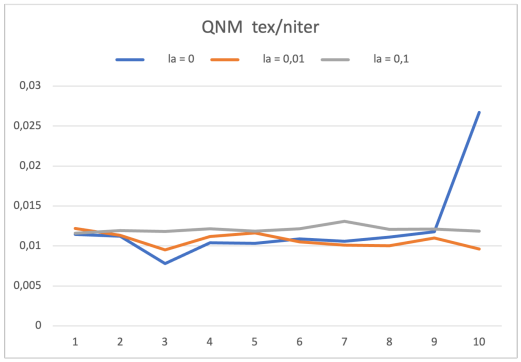
Table 8: Average number of iterations and execution time for each method.

This time the case is totally opposite to the previous, the SGM is the one with worst local convergence since it has the greater ratio tex/inter; although, still maintains the greater number of iterations and execution time. The GM is the one with best local convergence since its speed is ten time faster than the QNM.

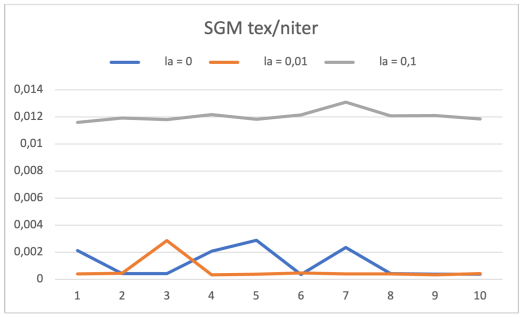
To summarize the previous representations, we plot the graphics of the ratio tex/niter to visually analyse the local convergence for each lambda value and each method.



(a) GM



(b) QNM



(c) SGM

Figure 5: Execution time and number of iterations ratio for each λ, target and method.

First, we analyse for each method which lambda makes the method the fastest possible. The GM has better local convergence when λ=0.01; the QNM when λ=0.01 (although in this case, all values of λ provide similar speeds); and the SGM when λ=0.01. With this results we are able to confirm that the value of lambda that makes all methods the fastest possible (more local convergence) is λ=0.01. As for which method is better in terms of local convergence, it seems that when differentiating lambdas the GM is better but in average is the SGM the one that is faster.

2.3 Discussion of Convergence Study

To determine which method is better in terms of global and local convergence we need to revise what we have studied. In the first step, we have checked for all three methods if they were globally

convergent and concluded that the method that is always global convergent and has less number of iterations is the QNM. Secondly, we have assigned the best method in terms of smallest value of loss function $\tilde{L}$ to each lambda and concluded that the global convergence does not depend on the regularization parameter. Thirdly, we have studied the execution time and concluded that the higher the λ the smaller the execution time, as well as the number of iterations. In the forth step, we have determined that the method with smaller average number of iterations and execution time is the QNM. In the last step, we have assigned the method with best local convergence to each λ value and concluded that GM and SGM have very similar behaviour but that GM is a bit better.

In conclusion, taking into account all these previous steps, it is sure to assume that the method that provides best global and local convergence is the Gradient Method. However, this method is not globally convergent when identifying certain target numbers, so it is not completely reliable; moreover, the SGM has very similar results and it does globally converge for all target numbers. Hence, we can affirm that the best method for identifying numbers in a sequence of digits is the Stochastic Gradient Method. As for the lambda, the one that gets the best results on both convergences is λ=0.01.

3 Study of recognition accuracy

Now we are going to analyse the recognition accuracy of SLNN, by checking the parameter *Accuracy TE*. We will run the training process for the ten digits and every algorithm GM, QNM, SGM using for all of them the lambda value that provides best accuracy, generated by the previous data.

In the first place we will study which lambda has the best accuracy over each method. For that, we represent in a table the accuracy values for each target and method together with its average for each λ.

Case λ=0:

λ=0.0			
target	**GM**	**QNM**	**SGM**
1	100	100	100
2	99,2	99,2	99,2
3	95,2	98	98
4	100	100	100
5	100	99,6	100
6	98	98,4	99,2
7	99,6	100	100
8	97,2	98,8	97,6
9	96,8	98,4	98,8
0	99,2	99,6	99,2
average	**98,5**	**99,2**	**99,2**

Table 9: Accuracy of each method in data set 1 with λ=0.

Case λ=0.01:

λ=0.01			
target	**GM**	**QNM**	**SGM**
1	100	100	100
2	99,2	99,2	99,2
3	98,8	98,8	98,8
4	100	100	100
5	100	100	100
6	99,6	99,6	99,6
7	100	100	100
8	98,8	98,8	70
9	99,2	99,2	99,2
0	100	100	99,6
average	**99,56**	**99,56**	**96,64**

Table 10: Accuracy of each method data set 1 with λ=0.01.

Note that GM and QNM have exactly the same accuracy values for each target number.

Case λ=0.1:

λ=0.1			
1	100	100	93,6
2	93,6	93,6	92
3	98	98	89,6
4	100	100	100
5	100	100	97,6
6	100	100	90,8
7	100	100	99,6
8	83,6	83,6	13,6
9	98,8	98,8	88,4
0	99,2	99,2	92
average	**97,3**	**97,3**	**85,7**

Table 11: Accuracy of each method data set 1 with λ=0.1.

These tables show the accuracy of all target numbers with respect to the three lambda values and the three methods, to summarize we have calculated their average. So, as for the GM we see that the best accuracy average is 99.56 given by λ=0.01. For QNM the best accuracy average is 99.56 given by λ=0.01, although its value is close to the one obtained with lambda 0. For SGM the best accuracy average is 99.2 given by λ=0. For all methods we obtain accuracy superior to 99%.

3.1 Best Method analysis

In this part we perform the execution of the script 'uo nn batch' in order to solve the problem in a bigger scale: 20000 samples. Taking into account the previous considerations on lambda value

for each method. With this generation of a new data set we will analyse the results and pose the following the question, is there a method that clearly outperforms the others in terms of training speed and recognition accuracy?

Once this new data set is generated we can calculate the average execution time, number of iterations, their ratio and also, add the accuracy of each method with their assigned lambda values. All this is shown in the next table:

	Average niter	**Average tex (s)**	**Average tex/niter (s)**	**Accuracy**
GM (λ=0.01)	83,2	249,4113	3,439	98,98
QNM (λ=0.01)	42,1	187,588	4,491	98,98
SGM (λ=0)	36391	4765,079	0,139	99,64

Table 12: Average number of iterations, execution time and accuracy for each method.

The first three measures are used to calculate the training speed of each method, the last one is for recognition accuracy; with both of them combined we have enough information to be able to determine which method is better. The information provided by the table is that the fastest method in terms of number of iterations is QNM, the one that reaches the optimal solution in less execution time is again QNM; despite that, the one that has better ratio *tex/niter* is SGM, this is the true valuable measure the one that has more speed per iteration. These results match the previous study with the first data set. As for the recognition accuracy the best method is SGM, although, there is a slight difference this is the best one.

In order to better visualize the accuracy of each method with the lambda that best performs, we have plot a chart.

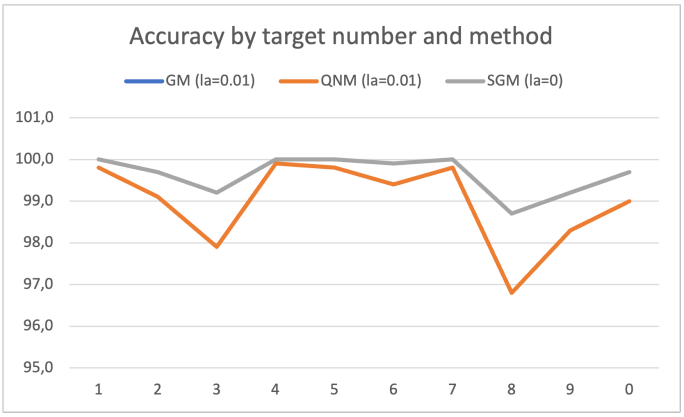


Figure 6: Accuracy of the three models with their corresponding lambdas.

The visualization only shows two lines, that is because the GM and the QNM have exactly the same accuracy values for all targets, something that we have also seen in the previous tables. Then again,

we determine that the SGM is the best method in terms of accuracy, in all targets the accuracy level is superior to the GM and QNM.

3.2 Discussion of $Accuracy^{TE} - \tilde{L}$ with respect to λ-algorithm

In this part we will see if the best method in terms of *Accuracy TE* coincides with the best on in terms of minimization of the loss function $\tilde{L}$.

As seen on the part 1, the method that minimizes the loss function $\tilde{L}$ is the SGM; this result also makes sense with the analysis of this second part. This time, though, is with lambda 0.

On the one hand, we have see that the lambda that gives the best convergence is λ=0.01, so if our goal is to minimize the loss function $\tilde{L}$ then we should choose this value. If our goal is to obtain the best accuracy we should choose λ=0, that provides accuracy of the 99.64%. In any case the best method is the SGM. On the other hand, if the methods desired are Gradient and Quasi-Newton then the lambda used to obtain best results should be λ=0.01, which provides 98.98% of accuracy.

In conclusion, both batch codes with 250 and 20000 samples work at their best with the same method, the Stochastic Gradient. It provides the best accuracy, the best ratio between number of iteration and execution time to do the best digit number recognition.