Optimization Techniques for Data Mining: Unconstrained Optimization

Pietro Fronte, Carolina Jorge Centeio

October 2018

1 Introduction

The assignment was about building a shallow Neural Network (just input layer and output layer) and train it with different dataset and optimization algorithms studied during the Unconstrained Optimization part of the course:

1. Steepest Descent Gradient
2. Conjugate Gradient Fletcher-Reeves
3. Conjugate Gradient Polak-Ribière
4. Quasi-Newton method BFGS
5. Quasi-Newton method DFP

The values of two input independent variable X_1 and X_2 are fed into the two neurons of the input layer. Each neuron is equipped with the Sigmoid function as activation function defined as follow:

$$o_i = \sigma(x_i) = \frac{1}{1 + e^{-X_i}}, \quad i = 1, 2$$

The values coming out from the neurons are now values between 0 and 1. We can call them output values O = $\{o_1, o_2\}$.

These values are then weighted and added together to get a new input value

$$X_3 = \sum_{i=1}^{2} w_i * O_i = w_1 * o_1 + w_2 * o_2$$

Input value passed to the third neuron, the only one belonging to the output layer. Again the Sigmoid function is applied to get the final output value y.

$$y(x, w) = \sigma(X_3) = \frac{1}{1 + e^{-X_3}} = \frac{1}{1 + e^{-\sum_{i=1}^{2} w_i * O_i}}$$

As a usual ML problem we need a function to measure the goodness of out classification, in this case we will use the Loss function for the training dataset defined as follow:

$$L(X^{TR}, Y^{TR}) = \min_{w \epsilon \mathbb{R}^2} L(w; X^{TR}, Y^{TR}) = \sum_{j=1}^{p} (y(X^{TR}, w) - Y^{TR})^2$$

With p referring to the dimension of the vectors X_1 and X_2

$$X_1 = x_{11}, x_{12}, ...x_{1p}, \quad X_2 = x_{21}, x_{22}, ...x_{2p}$$

The loss function will be our objective function to be optimized with respect to the parameter w. In order to increase the convexity of the function and facilitate the convergence of the algorithms we modify slightly the objective function adding a new parameter (not to be optimized):

$$\tilde{L}(w; X^{TR}, Y^{TR}, \lambda) = L(X^{TR}, Y^{TR}) + \lambda \frac{||w||^2}{2}$$

This is the true objective function we use, called Loss function with L_2 regularization with parameter λ.

2 First part

The first part of the assignment required the training of the NN on a training dataset built as follow

$$x_i \epsilon \{-1, 1\}, \quad y = \begin{cases} 1 \text{ if } x_2 = x_1 \\ 0 \text{ otherwise} \end{cases} \qquad i = 1, 2$$

Notice that when building the training dataset X^{TR} we are also building the response for each training datapoint Y^{TR}.

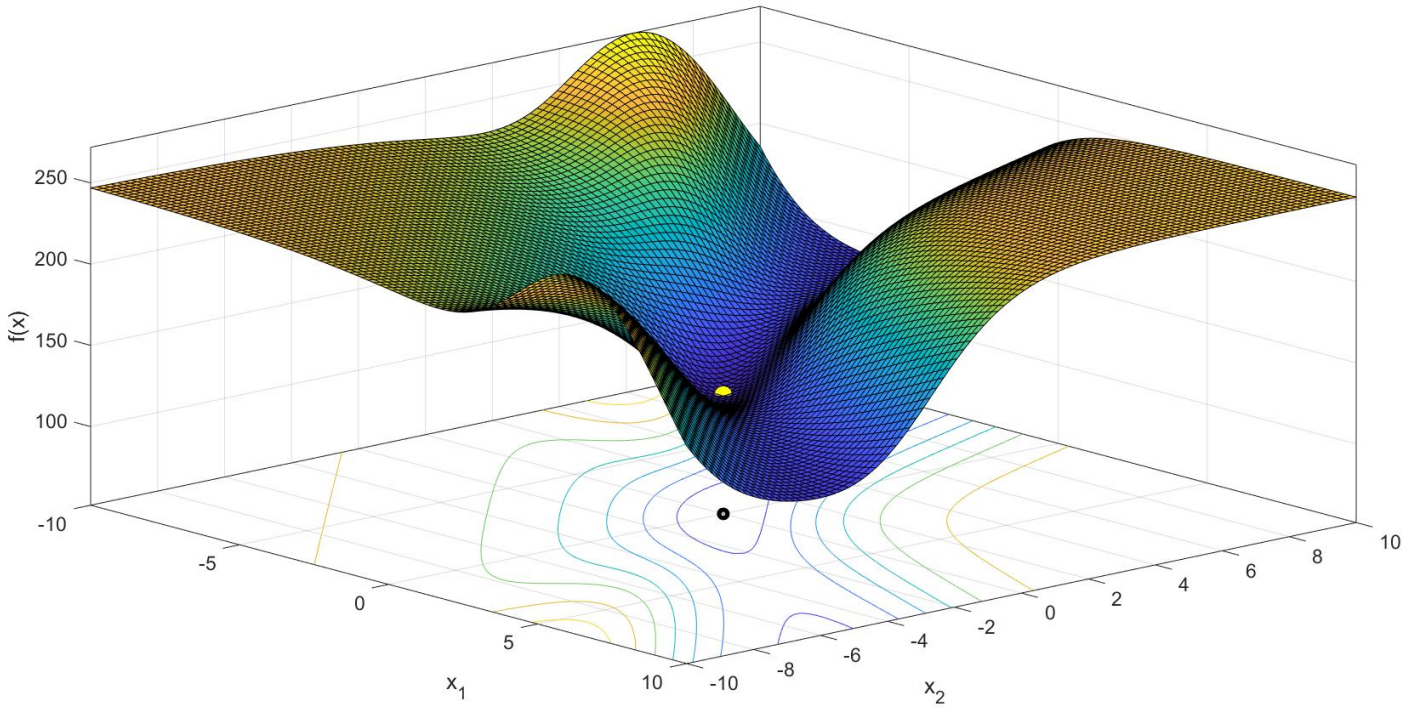
Notice that all the attempts done follow this rules:

1. size training dataset: 500
2. seed training dataset : 1234566
3. size test dataset: 50000
4. seed test dataset: 7891016
5. starting point w_0={0,0}

Starting with a λ=0 (then without regularization of the objective function) the results obtained are the following:

r	tr_p	tr_seed	la	w1(1)	w1(2)	ls	c2	sdm	iout	iter	w*(1)	w*(2)	L*	tr_acc	te_acc	te_q	te_seed
2	500	1234566	0.000	+0.00e+00	+0.00e+00	1	0.5	GM	0	13	-4.40e-03	-4.40e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	1	0.5	CGM-FR	3	35	-4.14e-03	-4.14e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	2	0.5	CGM-FR	0	3	+5.59e+02	+5.59e+02	2.520e+02	49.6	49.6	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	2	0.9	CGM-FR	0	3	+5.59e+02	+5.59e+02	2.520e+02	49.6	49.6	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	1	0.5	CGM-PR	2	2	-4.79e-03	-4.79e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	2	0.5	CGM-PR	2	2	-4.40e-03	-4.40e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	1	0.5	BFGS	0	7	-4.40e-03	-4.40e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.000	+0.00e+00	+0.00e+00	1	0.5	DFP	0	8	-4.40e-03	-4.40e-03	1.250e+02	50.4	50.4	50000	7891016

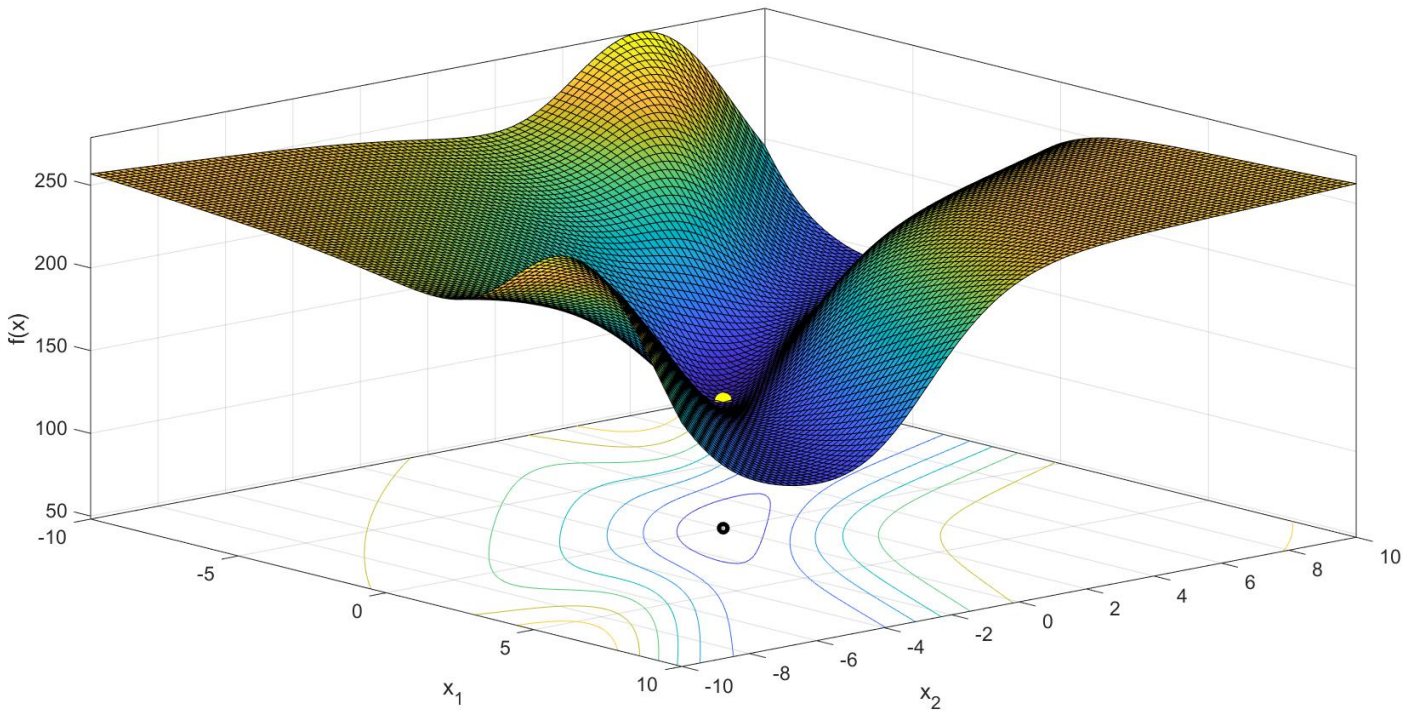
Figure 1: Function Optimization with λ=0



Moving to λ=0.1 the results are:

r	tr_p	tr_seed	la	w1(1)	w1(2)	ls	c2	sdm	iout	iter	w*(1)	w*(2)	L*	tr_acc	te_acc	te_q	te_seed
2	500	1234566	0.100	+0.00e+00	+0.00e+00	1	0.5	GM	1	101	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	2	0.5	GM	0	3	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	1	0.5	CGM-FR	1	101	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	2	0.5	CGM-FR	0	3	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	1	0.5	CGM-PR	2	2	-4.79e-03	-4.79e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	2	0.5	CGM-PR	2	2	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	1	0.5	BFGS	0	16	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	2	0.5	BFGS	0	3	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	1	0.5	DFP	0	9	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016
2	500	1234566	0.100	+0.00e+00	+0.00e+00	2	0.5	DFP	0	3	-4.39e-03	-4.39e-03	1.250e+02	50.4	50.4	50000	7891016

Figure 2: Function Optimization with λ=0.1



2.1 Comments of first results

In the not normalized problem there was no Conjugate Gradient Method able to carry out the optimization problem (FR in one case didn't converge and in the other case found the optimal in a really unusual tuple of w – PR was unable to converge with both linesearch algorithms). BFGS performed better then the others in this task.

The normalized problem show some anomalies:

1. Gradient method is not able to converge in 100 iteration but converges in only 3 iterations with the one-dimensional minizer instead of Backtracking linesearch
2. Same behaviour as Gradient method for Conjugate FR
3. Conjugate PR catch the optimal values without converging

Again the Quasi-Newton algorithms, together with the Steepest Descent performed better then the others getting to the optimal in only 3 steps.

3 Second part

Since in the first part of the Assignment no one of the algorithm used manage to catch an acceptable value of test accuracy (a 50% in test accuracy it means that it is just a random classification) we now try to add more information in the input variables hoping to get better results.

The training (and test) dataset this time is built as follow:

$$x_i \epsilon \{-1, 1\}, \quad x_3 = x_2 * x_1, \quad y = \begin{cases} 1 \text{ if } x_2{=}x_1 \\ 0 \text{ otherwise} \end{cases} \qquad i = 1, 2, 3$$

This new problem leads to some change in our previous set-up. Now we have:

$$o_i = \sigma(x_i) = \frac{1}{1 + e^{-X_i}}, \quad O = \{o_1, o_2, o_3\}$$

And the new input variable of the output layer will be:

$$X_4 = \sum_{i=1}^{3} w_i * O_i = w_1 * o_1 + w_2 * o_2 + w_3 * o_3$$

Keeping with the same objective function and algorithms the results obtained are:

Without regularization $\lambda = 0$

r	tr_p	tr_seed	la	w1(1)	w1(2)	w1(3)	ls	c2	sdm	iout	iter	w*(1)	w*(2)	w*(3)	L*	tr_acc	te_acc	te_q	te_seed
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	GM	0	424	-2.73e+01	-2.74e+01	+6.72e+01	3.602e-06	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	CGM-FR	2	9	-3.74e+00	-3.77e+00	+9.79e+00	1.857e+01	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	CGM-FR	0	68	-2.78e+01	-2.79e+01	+6.85e+01	2.556e-06	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	CGM-PR	2	6	-5.93e+00	-5.70e+00	+1.39e+01	6.148e+00	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	CGM-PR	2	31	-1.28e+01	-1.29e+01	+3.17e+01	6.164e-02	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	BFGS	0	12	-5.72e+01	-4.71e+01	+1.32e+02	4.716e-11	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	BFGS	0	11	-5.77e+03	-3.83e+03	+9.68e+03	0.000e+00	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	DFP	0	22	-3.63e+02	-1.97e+02	+7.37e+02	3.923e-36	100.0	100.0	50000	7891016
21	500	1234566	0.000	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	DFP	0	10	-3.92e+01	-3.26e+01	+8.74e+01	6.285e-08	100.0	100.0	50000	7891016

Adding regularization term $= 0.1$

r	tr_p	tr_seed	la	w1(1)	w1(2)	w1(3)	ls	c2	sdm	iout	iter	w*(1)	w*(2)	w*(3)	L*	tr_acc	te_acc	te_q	te_seed
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	GM	0	248	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	CGM-FR	0	111	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	CGM-FR	0	61	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	CGM-PR	2	6	-5.09e+00	-4.90e+00	+1.24e+01	1.934e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	BFGS	0	15	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	BFGS	0	14	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	1	0.5	DFP	0	16	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016
21	500	1234566	0.100	+0.00e+00	+0.00e+00	+0.00e+00	2	0.5	DFP	0	10	-5.36e+00	-5.43e+00	+1.31e+01	1.894e+01	100.0	100.0	50000	7891016

Due to the increased dimensionality of the problem is no more possible to plot it.

3.1 Comments on second results

The first thing that pops up looking at the table is that there is no optimal solution w:w_1,w_2,w_3 equals to any other in the table. Almost all the algorithms catch the optimal solution finally getting a 100% accuracy in the test set but there is no universal optimal solution

Adding convexity to the problem (moving λ to 0.1) we fixed this issue. Now (almost) all the algorithm are able to converge, find the optimal solution in w:-5.36, -5.43, 13.1 and get a 100% accuracy in test set. Again the Quasi-Newton show their performance against the competitors while the Conjugate PR didn't manage to converge in any problem even stopping in the optimal solution (see r2 with λ=0)