Mat II – Gradient Descend I Report

Sebastijan Trojer

April 7, 2025

1 Introduction

In this homework we implemented gradient descent and projected gradient descend. First we computed the function properties (smoothness (β), convexivity (α) and L factor). Then we used these parameters to evaluate approaches in python and compared them to theoretical guarantees.

2 Methodology and results

2.1 Task 5: PGD with different learning rates and domains.

Here we dealt with the function $f(x, y) = x^2 + e^x + y^2 - xy$, three different learning rates and three convex sets, which we restricted f to.

The α, β and L parameters we derived were the following (Table 1):

α	β	L
1.07	9.52	14.57

Table 1: Function parameters

We then tested three different learning rates, given in the following equations:

$$\gamma_1 = \frac{\|x_1 - x^\star\|}{L\sqrt{T}} \qquad \gamma_2 = \frac{1}{\beta} \qquad \gamma_{3_k} = \frac{2}{\alpha(k+1)} \tag{1}$$

And constrained f to the following domains:

$$\mathcal{D}_1 = \left\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1.5\right\} \tag{2}$$
$$\mathcal{D}_2 = [-1, 1] \times [-1, 1] \tag{3}$$
$$\mathcal{D}_3 = \operatorname{conv}\left\{(-1, -1),\ (1.5, -1),\ (-1, 1.5)\right\} \tag{4}$$

The initial point was $x_1 = (-1, 1)$, and the projections on the domains are derived on paper. For each combinations of the specified inputs and parameters we performed 10 steps of projected gradient descend. The results are shown in Table 2.

Domain	Learning rate	x	y	f(x, y)	Theoretical condition check
$\mathbb{R}^2$	N/A	-0.433	-0.216	0.789	N/A
$\mathcal{D}_1$	γ_1	-0.012	-0.444	1.179	$0.492 \leq 5.411$
	γ_2	-0.355	-0.084	0.804	$0.015 \leq 4.073$
	γ_{3_k}	**-0.438**	**-0.224**	**0.789**	$2.656 \times 10^{-5} \leq 36.569$
$\mathcal{D}_2$	γ_1	-0.005	-0.441	1.187	$0.919 \leq 6.228$
	γ_2	-0.355	-0.084	0.804	$0.015 \leq 5.406$
	γ_{3_k}	**-0.432**	**-0.217**	**0.789**	$0.003 \leq 36.569$
$\mathcal{D}_3$	γ_1	0.001	-0.347	1.121	$0.869 \leq 6.228$
	γ_2	-0.355	-0.084	0.804	$0.015 \leq 5.406$
	γ_{3_k}	**-0.483**	**-0.285**	**0.793**	$0.028 \leq 36.569$

Table 2: Points and function values (x, y, f(x,y)) at 10 steps of PGD. True minimum value of the function is 0.789 at point (-0.433, -0.21). The theoretical condition check contains computed constraints of corresponding claims in Theorem 3.3: 1. for γ_1, 2. for γ_2 and 4. for γ_{3_k}

Notice that all the theoretical conditions apply, and the conditions are very generous, in a sense that the actual values are way lower and the bounds aren't very strict, as they have to apply for the worst-case scenarios. All experiments achieved the function value difference significantly below the threshold. The best performing learning rate was γ_{3_k}, which might be due to it being an adaptive learning rate, which prevented overshooting. The domain with the best convergence was $\mathcal{D}_2$, with the other two also being very close. The same insight is provided by the theoretical values – the square and triangle domains have much lower values on the left sides of the equations than the circle. If we closely inspect the ending points on Figure 2.1, the distance of the points from the global minimum correspond to the results in this table. Another interesting insight is that the best performing descent on the square domain came closer to the global minimum, but the value at the bound is higher, which implies that the descend was smoother (the average x_i was closer to the global minimum), also seen on the trajectory figure.

Figure 2.1 shows the function $f(x, y)$, its level lines and the points of convergence for all the runs. Furthermore, on the right side it shows the trajectories of all the descents in different domains and different learning rates.

Notice that all the γ_2 trajectories overlap, since the learning rate is the exact same for all the domains, which also means that all the end points are exactly the same, since all the starting points are the same as well.

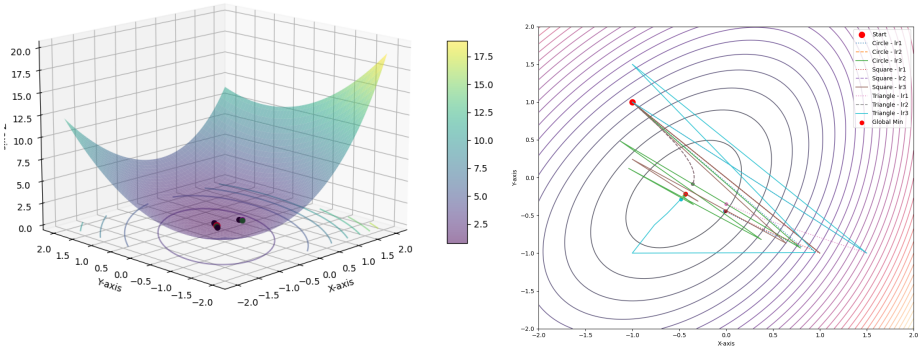

Figure 1: The left image shows the following: $f(x, y)$, level lines and end points from Table 2. The domains are colored green, pink and black respectively. The right image shows optimization trajectories for different domains and learning rates.

2.2 Task 6: Finding global minimum of a given function.

In this task we were tasked with finding the minimum of the following function:

$$f(z_1, z_2, z_3) = -\sum_{i=1}^{4} c_i e^{\left(-\left(\sum_{j=1}^{3} a_{i,j}(z_j - p_{i,j})^2\right)\right)}$$

defined on the domain $[0, 1]^3$ with values of the coefficients reported in the instructions. We performed up to 1000 steps of PGD with learning rate 0.01 with exponential decay (in our case we chose to reduce it by 1% each iteration). We also set up a convergence criteria, meaning we stopped the descend once the distance between x_{i-1} and x_i was less than 10^{-6}. The minimal value of f is approximately -3.8627.

We performed such PGD from 10 random points. PGD usually converged after between 300 and 700 iterations, and the closest we got to the global minimum was the function value -3.859, the absolute difference between function values being 0.0253. This specific convergence happened in 718 iterations. We also tried removing the convergence criteria lower and increasing the number of iterations, but we only achieved a marginal improvement, even after testing different learning rates and decays.

3 Conclusion

We tested multiple gradient descent approaches on 2 functions, defined projections on 3 different domains and analyzed the performance of 3 different learning rates. We visualized the descend process and provided insights into the effects of choosing each combination of parameters. This can serve as a reference when considering the approach when being tasked with minimizing a function.