Elia Fantini - Sciper n. 336006

Math of Data : Homework 2

Exercise 1.1.1|

Given $f: \mathbb{R}^2 \to \mathbb{R}$ such that $f(x,y) = (x-1)(y-1)$, we have that:

$f(x,y) = xy - x - y + 1$

The partial derivatives of f are:

$$\nabla_x f(x,y) = y-1 \qquad \nabla_y f(x,y) = x-1$$

Hence, the point $(x^*,y^*) = (1,1)$ is the only one for which the gradient of f $\nabla f = (\nabla_x f(x,y), \nabla_y f(x,y))$ is equal to zero. $(x^*,y^*) = (1,1)$ is then a first-order stationary point.

The Hessian of f is:

$$H_f = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Let's now calculate the characteristic equation of H_f to find its eigenvalues.

$$(H_f - \lambda I) = \begin{bmatrix} 0-\lambda & 1-0 \\ 1-0 & 0-\lambda \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 1 & -\lambda \end{bmatrix}$$

$$\det(H_f - \lambda I) = -\lambda \cdot (-\lambda) - (1 \cdot 1) = \lambda^2 - 1 \Rightarrow \lambda_1 = 1 \quad \lambda_2 = -1$$

From slide 8/53 of Lecture 9 we know that $(x^*,y^*) = (1,1)$ is then a saddle point.

Exercise 1.1.2|

Since $0 = f(1,1) \geq f(1,y) = 0 \cdot (y-1) = 0$ for all $y \in \mathbb{R}$

and $0 = f(1,1) \leq f(x,1) = (x-1) \cdot 0 = 0$ for all $x \in \mathbb{R}$

then $(x^*,y^*) = (1,1)$ is a solution to the minimax problem $\min_x \max_y f(x,y)$.

Exercise 1.1.3

Knowing $\nabla_x f(x,y)$ and $\nabla_y f(x,y)$, the equations for the simultaneous gradient descent/ascent updates become:

$$x_{k+1} = x_k - \gamma(y_k - 1)$$

$$y_{k+1} = y_k + \gamma(x_k - 1)$$

The sequence of iterates $\{x_k, y_k\}_{k=0}^{\infty}$ starting from any points $(x_0, y_0) \neq (1,1)$ and any $\gamma > 0$ if and only if the euclidean distance of (x_k, y_k) from $(1,1)$ $\|(x_k - 1, y_k - 1)\| \rightarrow +\infty$ for $k \rightarrow +\infty$.

$$\|(x_k - 1, y_k - 1)\| = \sqrt{(x_k - 1)^2 + (y_k - 1)^2}$$

$$= \sqrt{(x_{k-1} - \gamma(y_{k-1} - 1) - 1)^2 + (y_{k-1} + \gamma(x_{k-1} - 1) - 1)^2}$$

$$= \Big(x_{k-1}^2 - 2x_{k-1} + 1 + y_{k-1}^2 - 2y_{k-1} + 1 + \gamma^2(y_{k-1} - 1)^2 + \gamma^2(x_{k-1} - 1)^2$$
$$- 2x_{k-1}\gamma(y_{k-1} - 1) + 2y_{k-1}\gamma(x_{k-1} - 1) + 2\gamma(y_{k-1} - 1) - 2\gamma(x_{k-1} - 1)\Big)^{1/2}$$

$$= \left((x_{k-1} - 1)^2 + (y_{k-1} - 1)^2 + \gamma^2\left((y_k - 1)^2 + (x_k - 1)^2\right)\right)^{1/2}$$

$$= (\gamma^2 + 1)^{1/2}\left((x_{k-1} - 1)^2 + (y_{k-1} - 1)^2\right)^{1/2}$$

$$= (\gamma^2 + 1)^{1/2} \|(x_{k-1} - 1, y_{k-1} - 1)\|$$

$$\Rightarrow \frac{\|(x_k - 1, y_k - 1)\|}{\|(x_{k-1} - 1, y_{k-1} - 1)\|} = (\gamma^2 + 1)^{1/2}$$

Since $\gamma > 0$, $\sqrt{\gamma^2 + 1} > 1$, hence at each iteration the distance of points from the saddle point $(1,1)$ increases, so the sequence of iterates diverges with a rate of $(\gamma^2 + 1)^{k/2}$.

Exercise 1.2.2/1.2.3

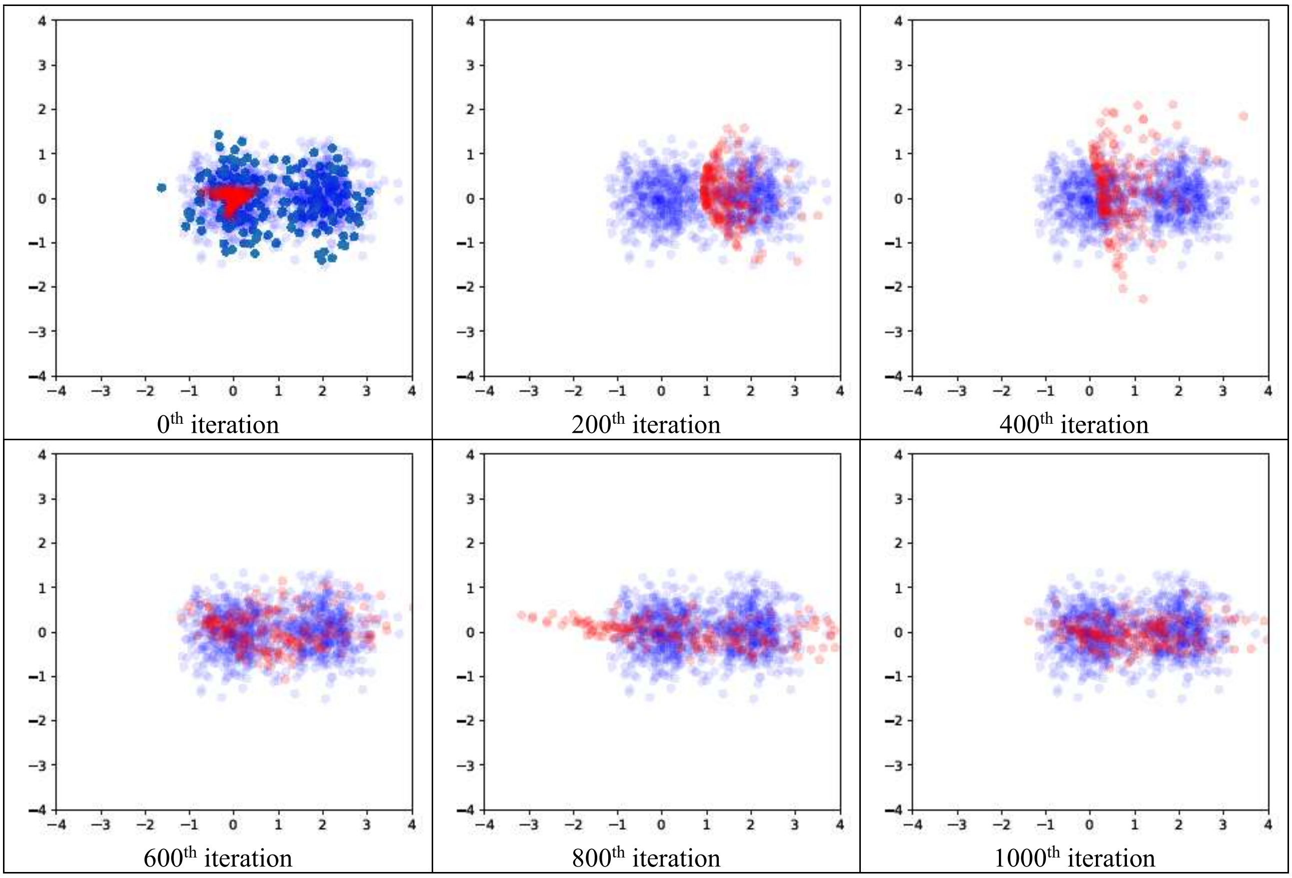
The use of Spectral Normalization to enforce the Lipschitz constraint presents different advantages compared to other techniques, and it is not computationally expensive thanks to the power iteration method that efficiently approximates the value of the largest singular value of the parameters' matrix.

Weight clipping, implemented with code in this homework, suffers from the same drawback of simple weights normalization with l2-norm: it reduces the rank of the matrix W and with it the number of features to be used for the discriminator. Due to this effect, only a few features will match the target distribution. On the contrary, the spectral norm is independent of rank as it only depends on the value of the maximum singular value, allowing W to use as many features as possible while satisfying the 1-Lipschitz constraint as well.

Gradient penalty's technique consists of adding to the objective function a regularizer that leads to a local 1-Lipschitz constant at discrete sets of points generated by interpolating a sample from generative distribution and one sample from the data distribution. The drawback here is that it depends on the support of the current generative distribution, which changes during the training and might decrease the performance of the training, especially with high learning rates. In addition to that, the gradient penalty proved to be more computationally expensive than spectral normalization as well.

In the end, spectral norm regularization is another technique based on the spectral norm. Unlike spectral normalization though, spectral regularization does not set the spectral norm to a designated value, it just adds a regularization term to the objective function that penalizes weights with a high spectral norm. Moreover, it imposes sample data-independent regularization on the cost function, whereas spectral normalization uses a sample data-dependent regularization function.

In the following figures (Figure 1, Figure 2) it has been reported how the behaviors of spectral normalization and weight clipping change in the implementation of this homework's problem over 1000 steps. From those results, we can see how with both techniques the noise (red points) tends to have the same distribution as real data (blue points) and they keep oscillating between a less precise solution and a more accurate one. Using spectral normalization makes the noise achieve a similar distribution to the real data one after 600 iterations, while weight clipping is faster and achieves it after about 300 iterations. Spectral normalization does also get worse after 800 iterations and then gets back to a good distribution after 1000, whereas weight clipping maintains a more constant result.



0th iteration

200th iteration

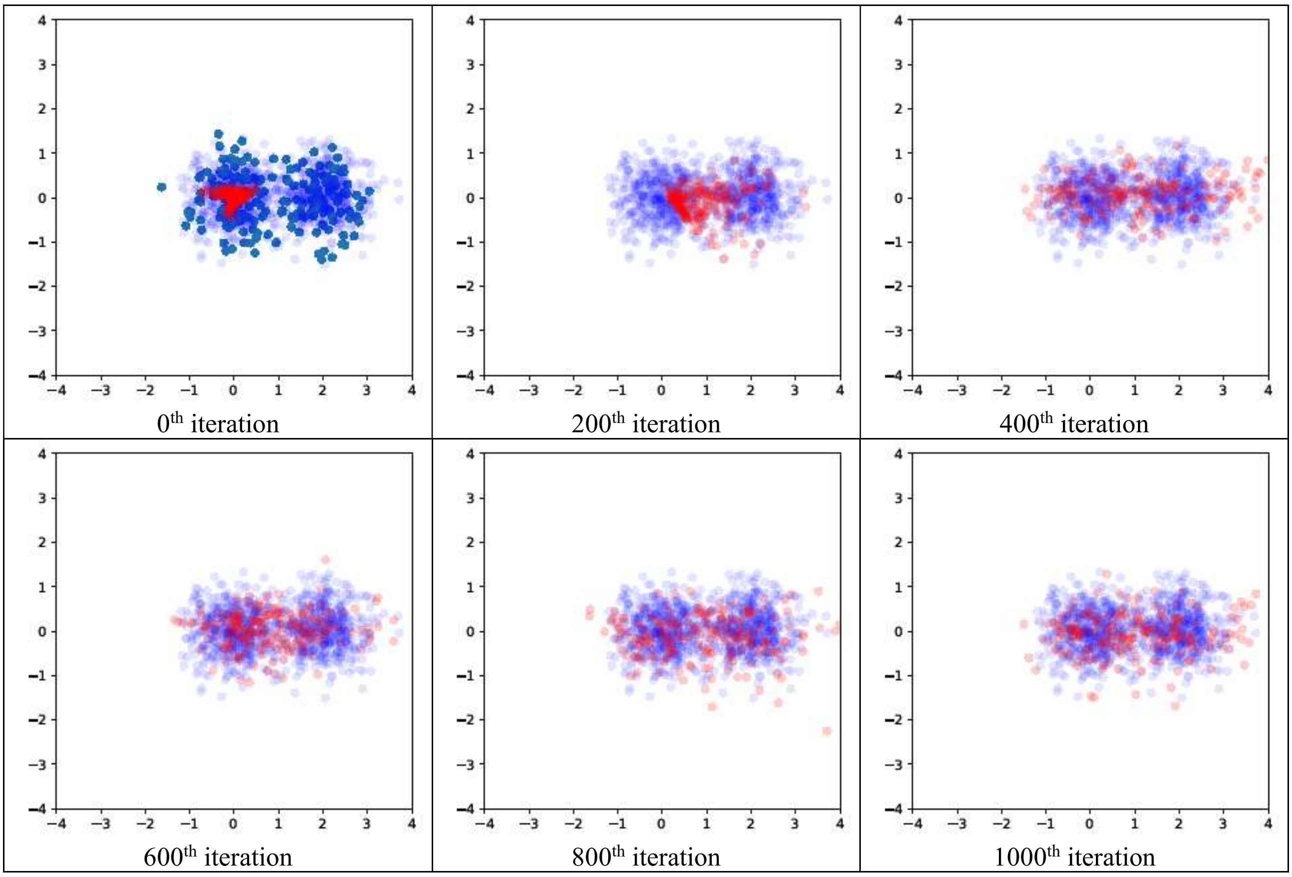
400th iteration

600th iteration

800th iteration

1000th iteration

Figure 1: Results with Spectral Normalization



0th iteration

200th iteration

400th iteration

600th iteration

800th iteration

1000th iteration

Figure 2: Results with Weight Clipping