Report Lab 1

Basic MLP Implementation with Numpy/PyTorch Using Optimizers and Autograd

preguntes:

* loss function 0.5? estaria millor fent la mitja, el 0.5 és per convenció per a q la derivada quedi més neta
* decision boundary
* report comentar epoches?
* can we use other libraries (sklearn) yes

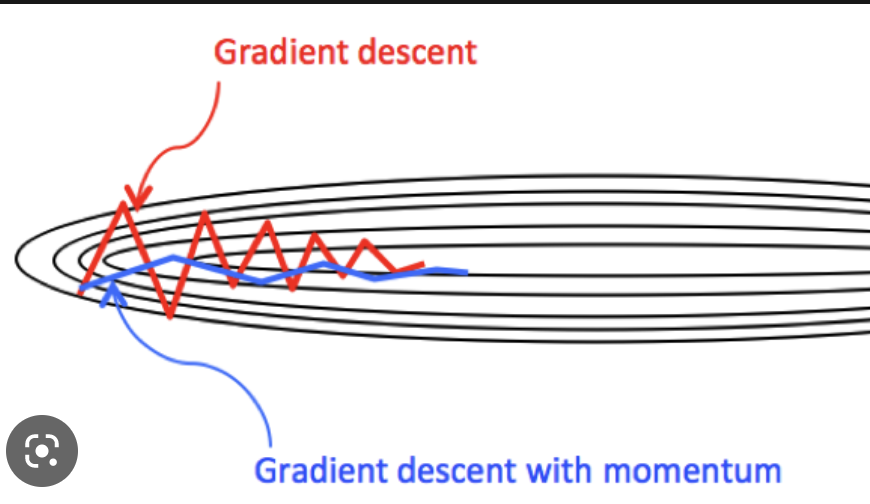
**Exercise 1.**

* **canviant epoches**

**Exercise 2.**

Implement momentum. Optimize learning rate so the descent does not oscillate.

* Reducing the learning rate does not help because in some dimension you are moving forward.
* We use the concept of momentum. We define that   
  zeta2 = zeta1 - alpha \* ( (gradient0 + gradient 1) / 2) by using this average we ensure that we do not oscillate in some dimension.



MOVING AVERAGE:

zetaT = zetaT-1 - alpha \* (1/T-K) \* SUM\_(i = T-K)^(T-1) (gradientI)

Momentum:

Vt = beta·Vt-1 + (1-Beta)\*alpha\*gradient(t), beta in [0,1]

zeta(t) = zeta(t-1) - Vt

**Exercise 3.**

questions: