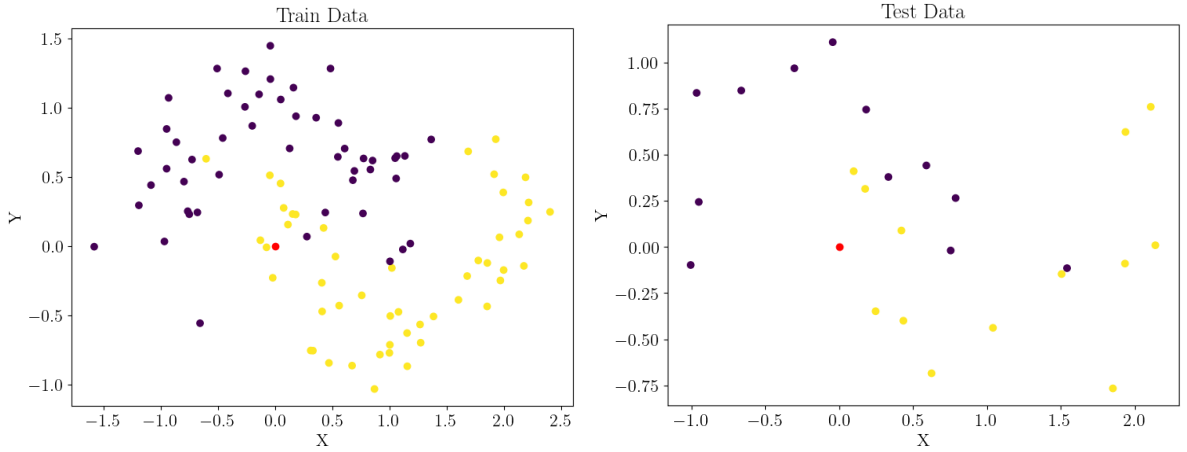
Sergio Pérez NIA: 242111// Albert Santos NIA: 240690// Edu Rodríguez NIA:242710

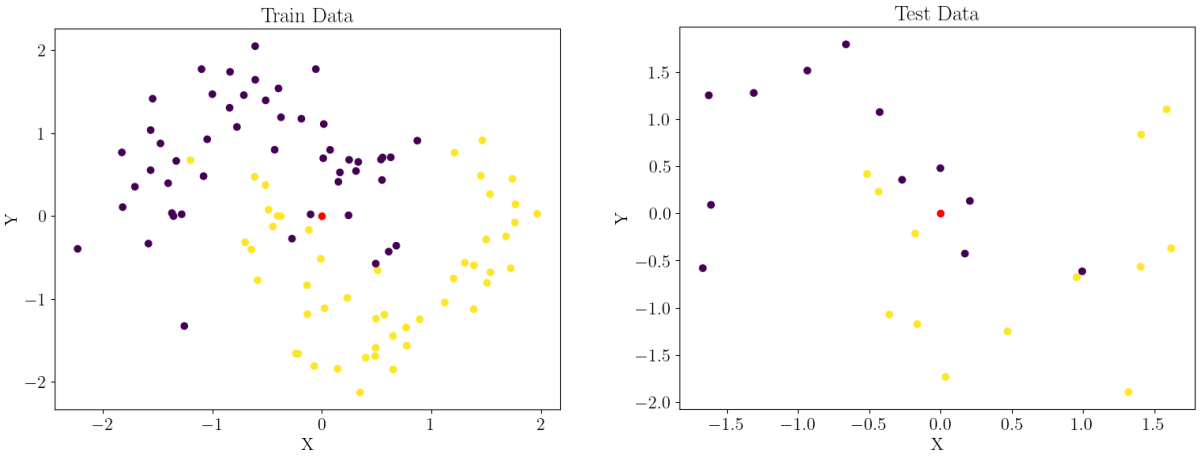
Report Lab1

EX1

Original Data



Normalised Data

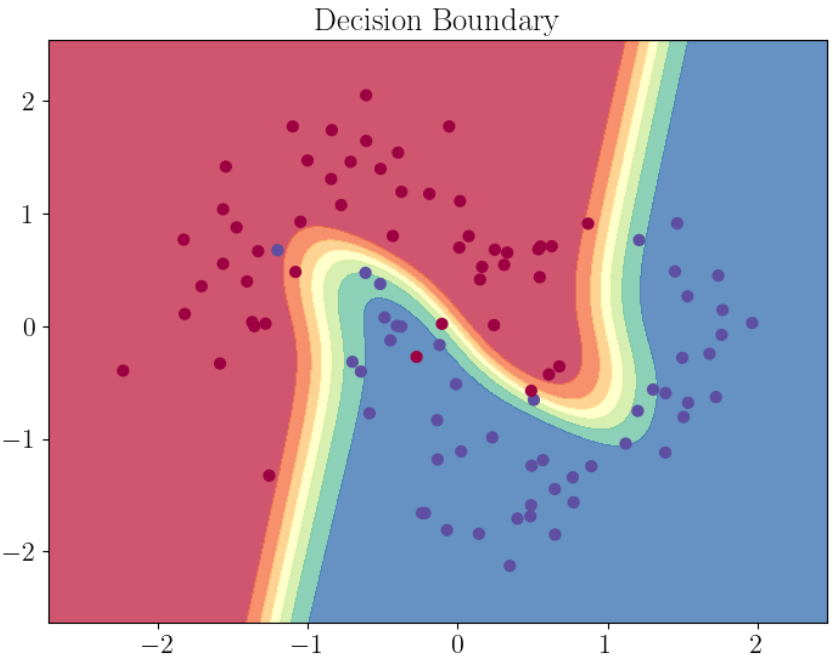


The red point in coordinates (0,0) helps us to see that data is correctly normalised.

In the multi-layer perceptron we have been told to change the loss function from Mean Square Error, to Cross Entropy loss. Bearing in mind that the MLP is implemented in numpy and has therefore no automatic derivation system, we will have to change the derivatives by writing them. Well, actually, it is only one derivative we have to change: the last in the forward propagation, or, equivalently, the first in the backpropagation. This is the one that computes the loss, and we will substitute it by the derivative of the cross entropy with respect to the $\hat{y}$, which I will express as a. Namely:

$$\frac{\partial L}{\partial a} = -\frac{y}{a} + \frac{(1-y)}{(1-a)}$$

This Decision boundary help us to see qualitatively,the zones for which our model will output 1 or 0 (in red or blue) and also the borders zones which could be possibles errors shown in others colours(like a gradient from red to blue)



To compute the accuracy we have of our Multi-Layer Perceptron we will consider the final output to be a probability of the variable belonging to class 1 such that if it is over 0.5 it is considered to predict class 1 and 0 otherwise.
By doing so we can compute the amount of accuracy simply by counting the number of occurrences where the prediction matches the label and then dividing by the number of total instances of the dataset being analysed.

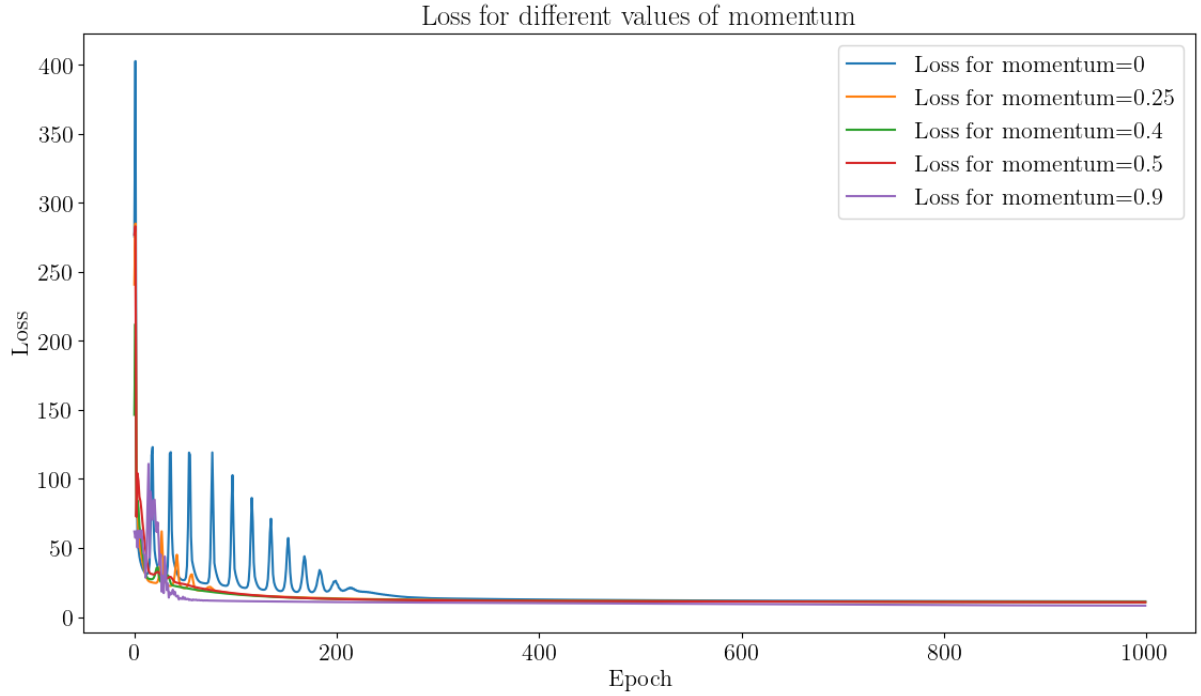
EX2

The objective of this exercise is to implement an optimizer which uses momentum, which will serve us to keep the previous direction of maximum decrease in the loss hyperplane and avoid oscillating unnecessarily. This optimizer is implemented by:

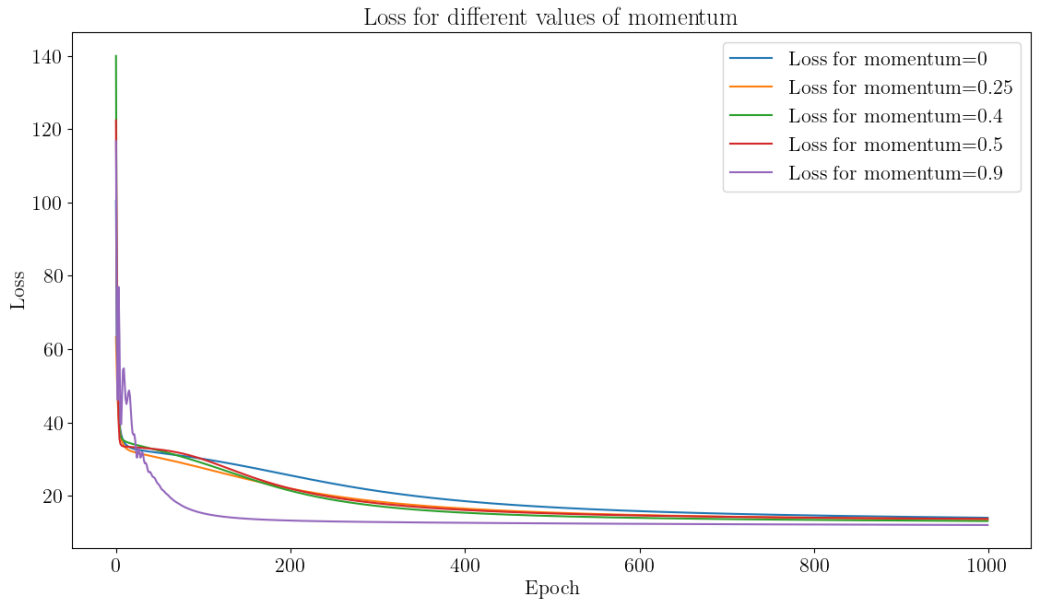
$$v_t = \gamma v_{t-1} + \eta \frac{\partial}{\partial \theta} J(\theta)$$
$$\theta = \theta - v_t$$

We try to train models with different momentums to see the difference between them. Momentum 0 will be like gradient descent.

In order to find the best hyperparameter configurations we have tried many combinations of layer size and learning rate, and have found that training a MLP of 128 nodes with a learning rate of 0.01 for 1000 epochs yielded great results quite quickly for all momentum values.



As the figure above depicts, the greater the parameter of the momentum the faster it gets to the minimum. Moreover, it is important to mention that the value helps to avoid overfitting: The blue peaks correspond to the SGD and demonstrate that the structure of the Neural network is too powerful for the task but if the momentum value is increased then the model is less susceptible to these peaks, without mentioning the 0.9. The figure above shows the how the model acts with 128 hidden neurons, if we plot the performance with 16 hidden neurons we get the following:

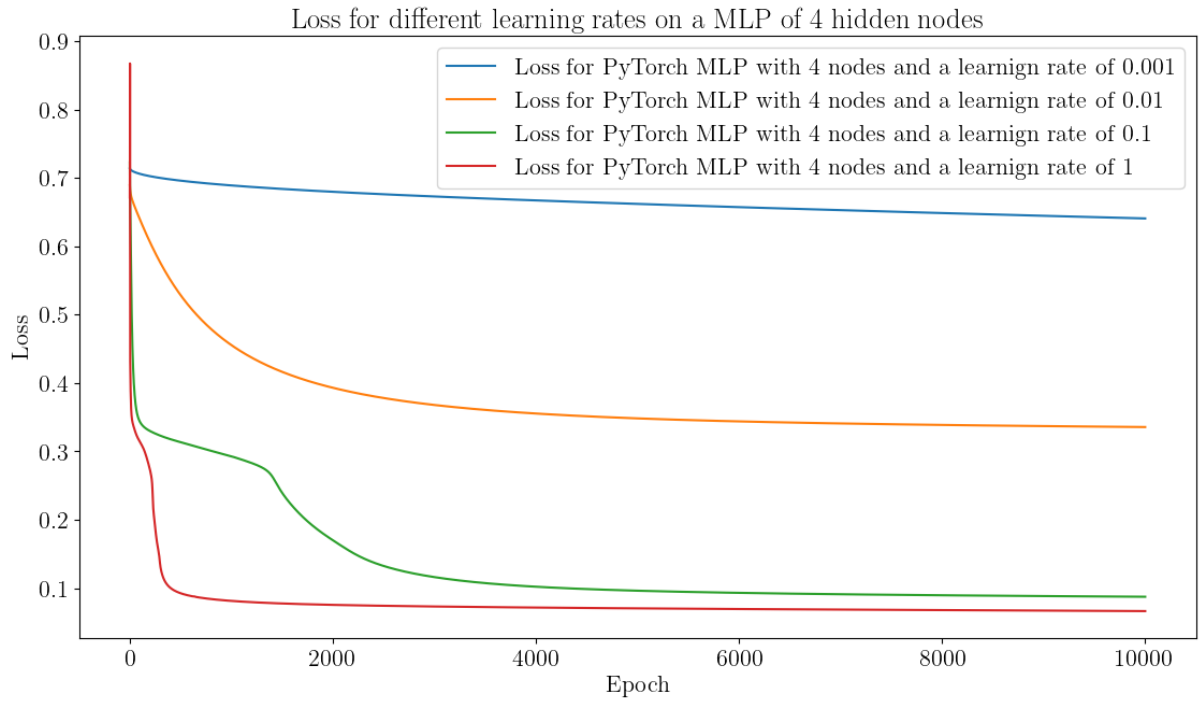


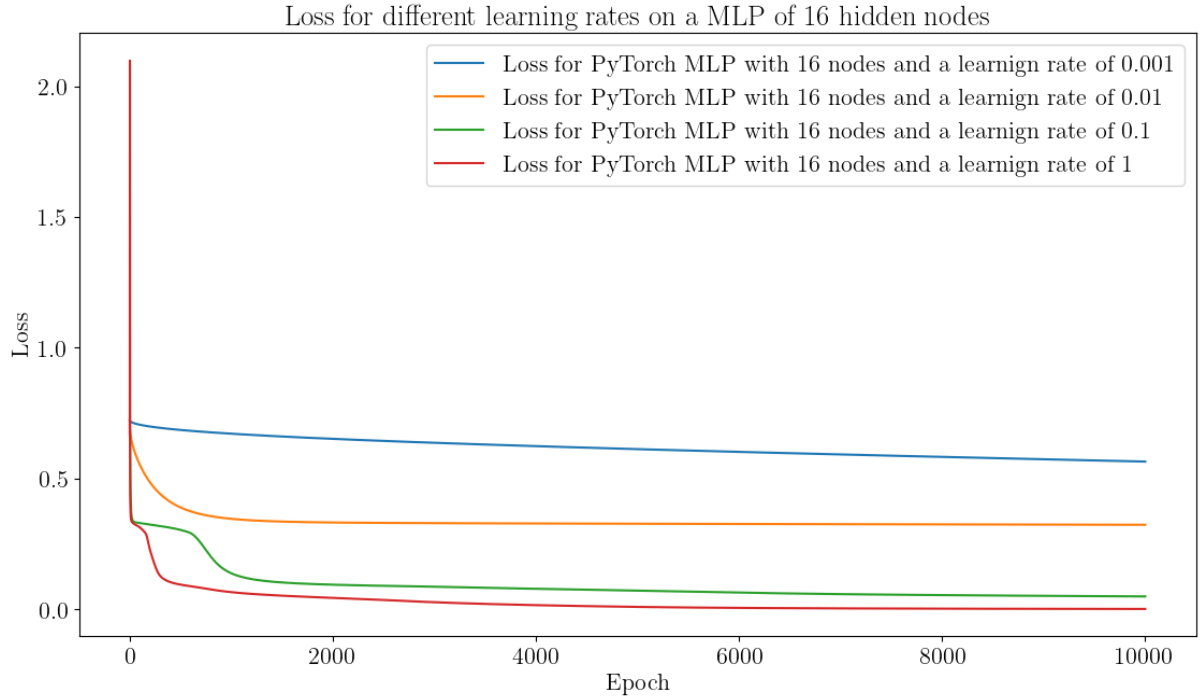
It is interesting to see again that the greater the value of the momentum parameter the faster it gets to the minimum. We only have plotted 1000 epochs but the loss gets decreased with the number of epochs. A great result will come with 100000 epochs.

EX3

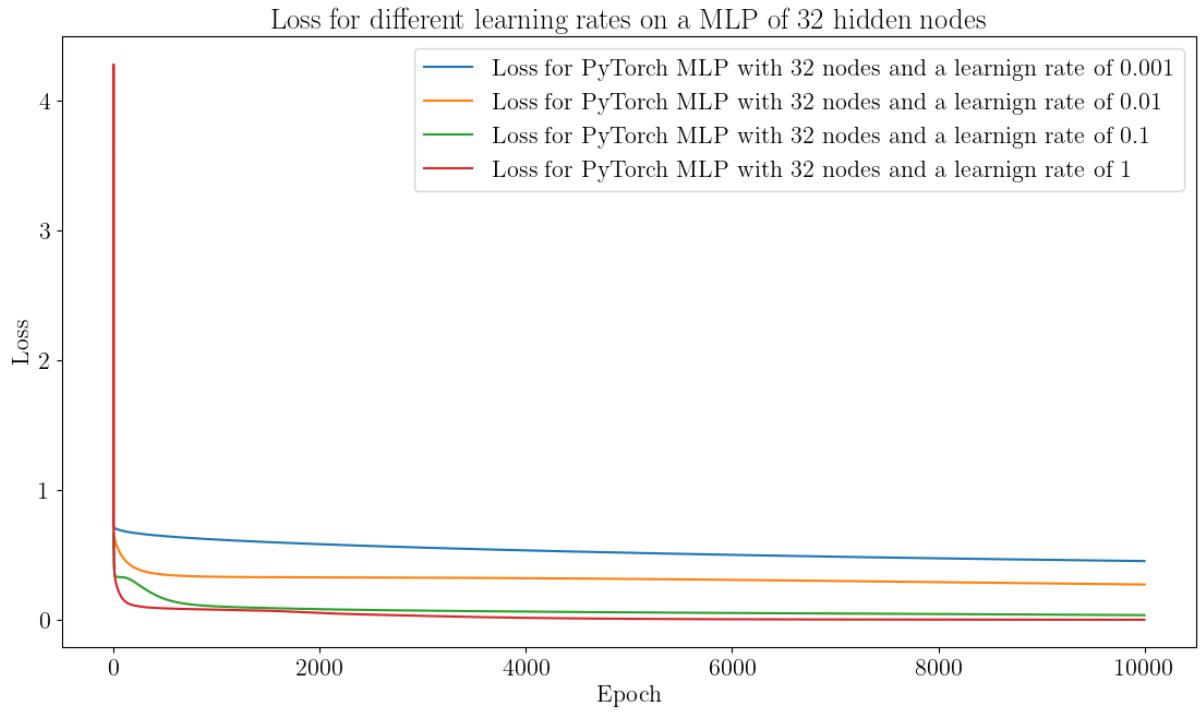
In this exercise we used pytorch to do a faster computation and to use its module functions. The model we created for this exercise in order to assess the same task as before needed the BCELoss function. Furthermore, we used the AdaGrad optimizer because it is an optimizer that adapts the learning rate of each parameter based on the historical gradient information. A great benefit of AdaGrad is that it requires very little tuning of hyperparameters. The only hyperparameter that needs to be tuned is the initial learning rate, which can be set to a small value. After that, the learning rate is adapted automatically for each parameter. This can be seen through the following plots, that with a great value of the learning rate the model does not even learn. Moreover, the number of hidden neurons is not as insightful as before due to the benefits of Adagrad.

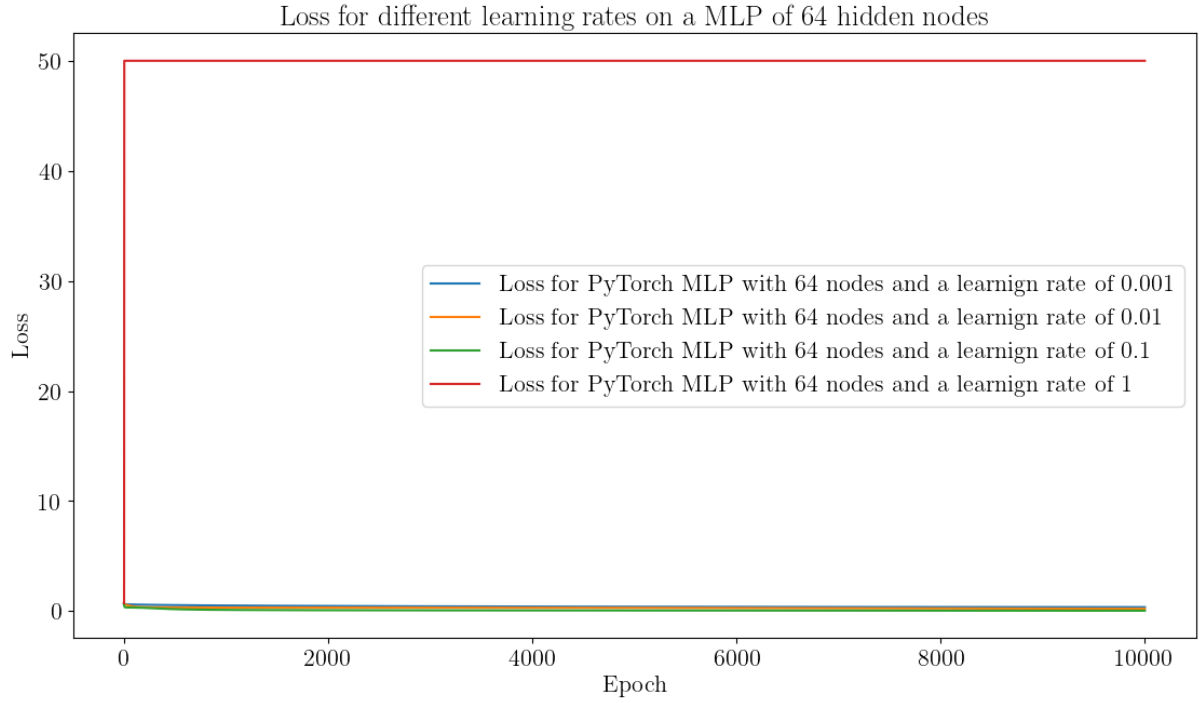
Sergio Pérez NIA: 242111// Albert Santos NIA: 240690// Edu Rodríguez NIA:242710



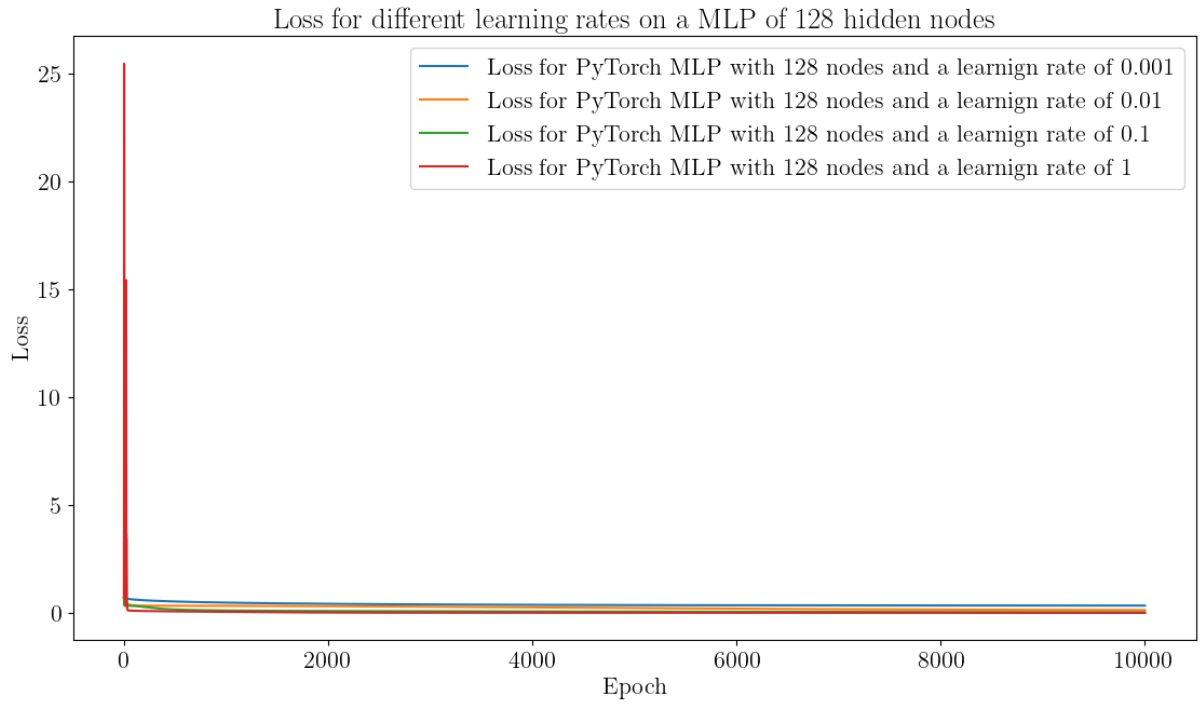


Sergio Pérez NIA: 242111// Albert Santos NIA: 240690// Edu Rodríguez NIA:242710





Sergio Pérez NIA: 242111// Albert Santos NIA: 240690// Edu Rodríguez NIA:242710



Sergio Pérez NIA: 242111// Albert Santos NIA: 240690// Edu Rodríguez NIA:242710