

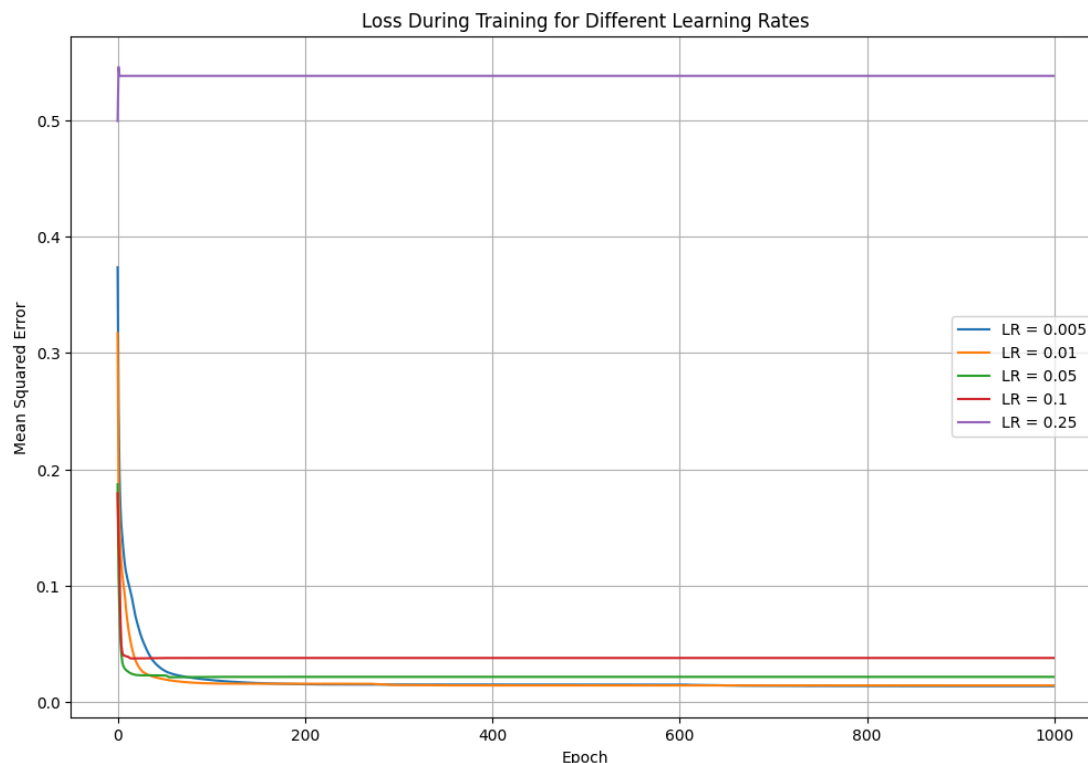
I have not found the Jupyter notebook that was built during the lectures so I wrote my own.

I have written the main neural network class based on Part X - Neural network mathematics in practice pdf file. The notebook is structured to be modular for easier answering of each question. The experiment functions serve as independent environments for each experiment.

Adjust the learning rate. Ranges between 0.005 and 0.25 are probably usable with this simple neural network. Does this affect the training process somehow?

This neural network is very basic in structure, and extreme values might break the learning process (which is probably also a good reason to just use TensorFlow, PyTorch? What do you think?)

Setting the learning rate from 0.005 to 0.1 has not shown significant effect but 0.25 has increased the mean error. So, I suppose we should not set this value too high.

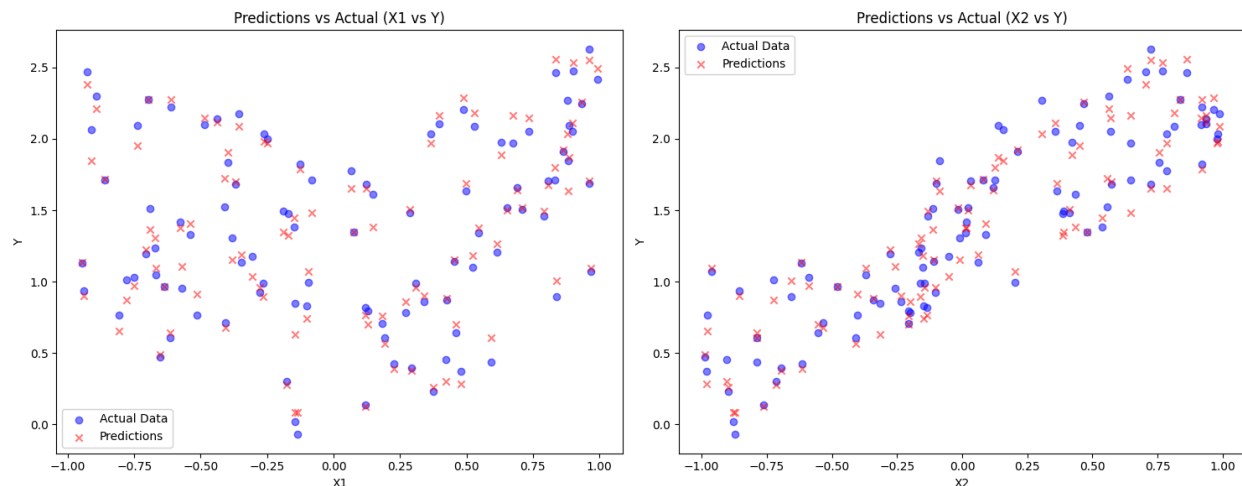


I think frameworks such as PyTorch or TensorFlow can simplify the development process and protect inexperienced users from noob making like setting some values to extreme. But experienced users or people with proper theoretical knowledge can be limited in their abilities due to design decisions made by libraries' authors, especially when it comes to performance. The decision on whether to use the libraries depends on a team's confidence in their both practical experience and theoretical knowledge behind machine learning.

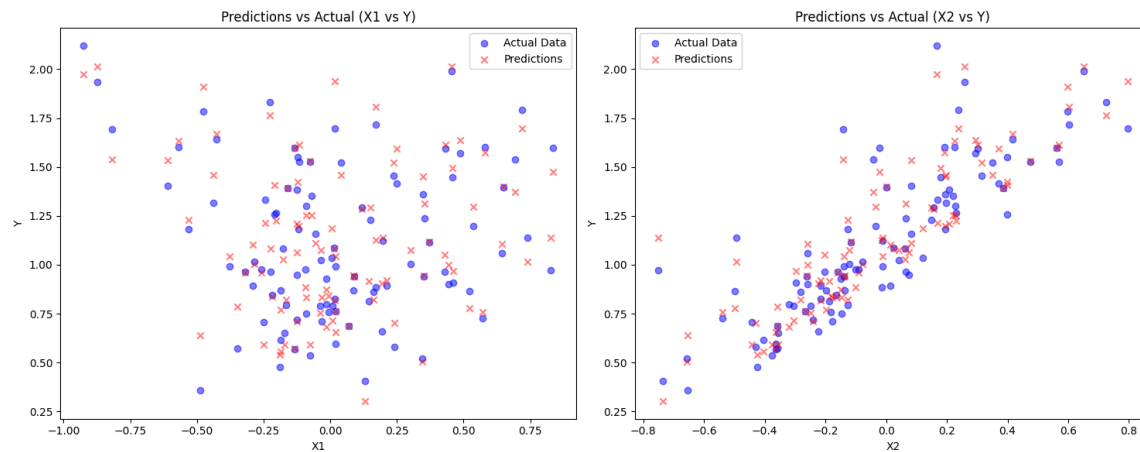
Play around with the data generator, try different ranges and different distributions of data. Does it affect the neural network's performance somehow?

I've run the same model with uniform and normal data distributions. Loss rate was better with normal distribution. But predictions worked visibly better with uniform data.

Predictions for uniform data:



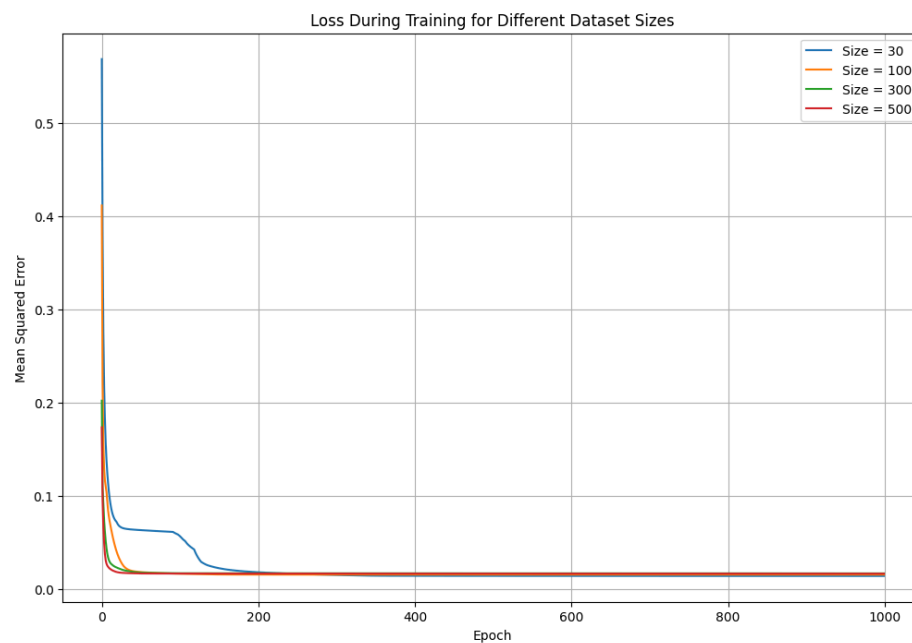
Predictions for normal data:



I'm not sure how much randomness played the role but the conclusion we get from these predictions is that different data distributions do affect neural networks' performance.

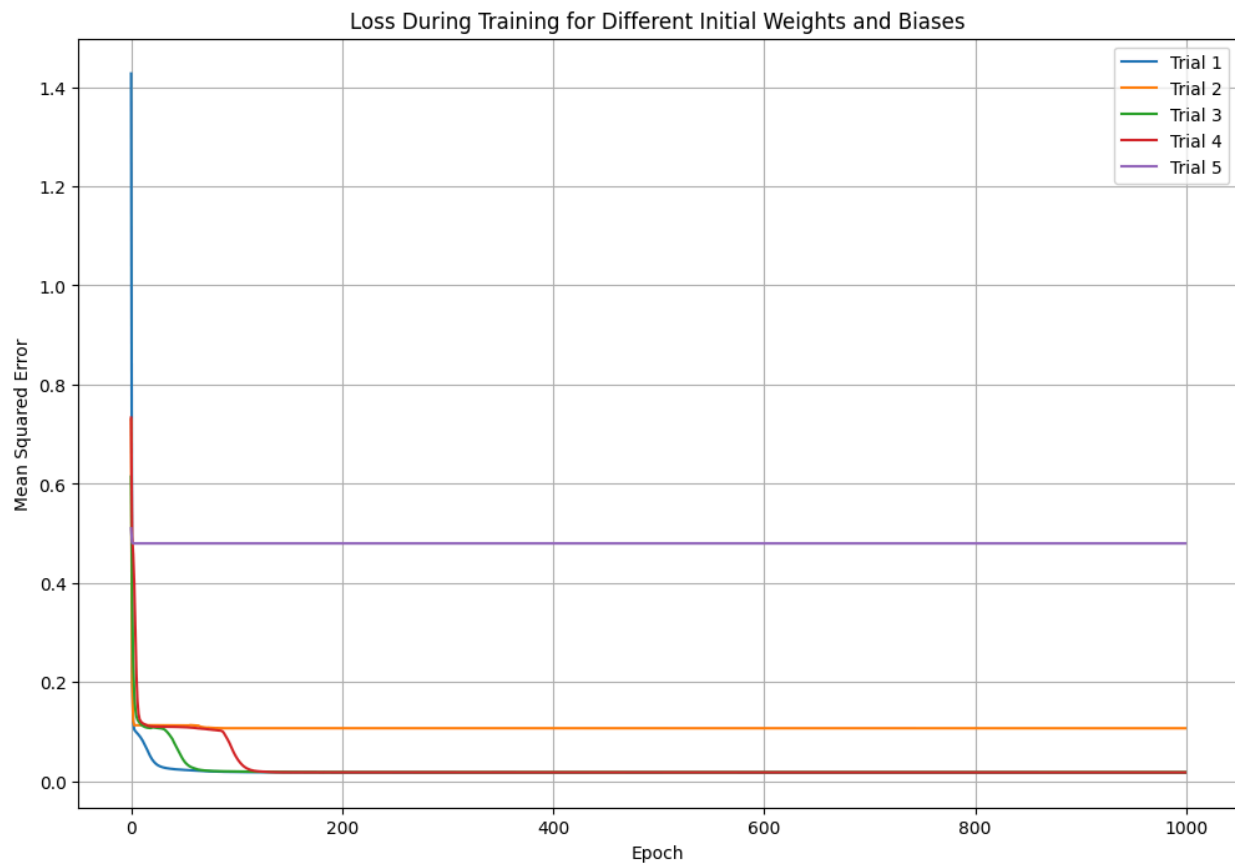
Try alternating the dataset size (30 – 500), does the performance change for the neural network?

I've tried training the network for 30, 100, 300, and 500 datasets. The results show that the bigger the size the better performance we get.



Try adjusting the starting weights and bias values a little. Is there an effect?

Random starting weights and bias did show different results.



Trial 5 has the biggest difference between others.

I recommend checking source files for better visibility of plots.