Assignment 1: Basics of Regression

UVA CS4774

February 14th, 2022

Owen Richards

A: Dataset:

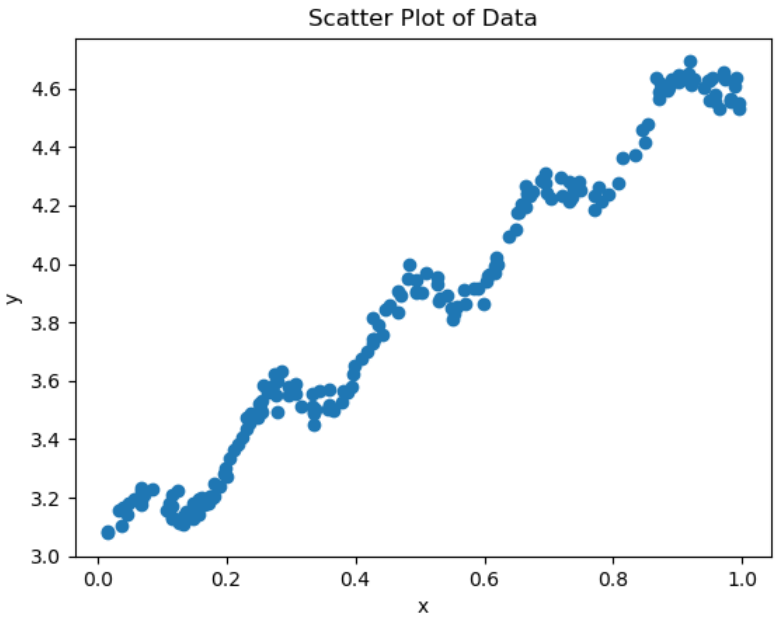

Fig 1: Plot of given dataset

B: Linear Regression with Normal Equation:

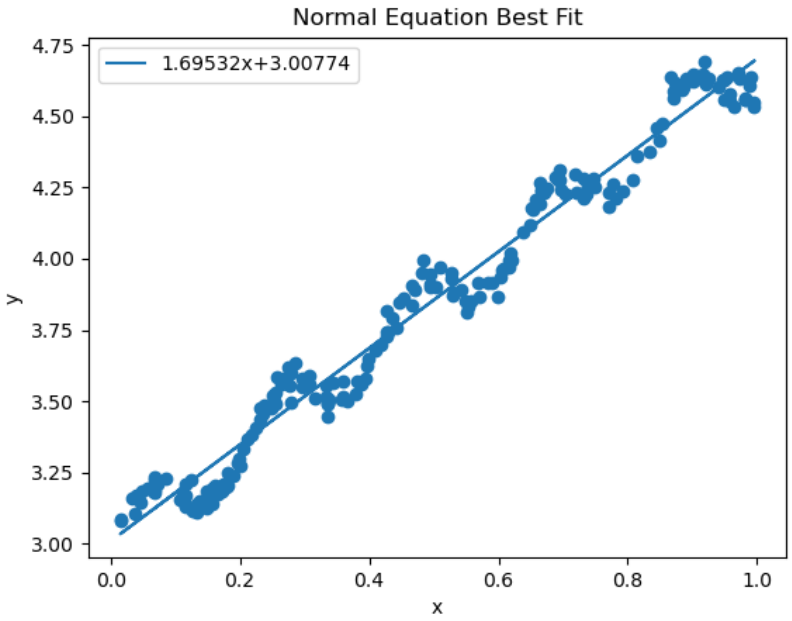

Fig 2: Best-fit LR line with the Normal equation

C: Line Regression with Gradient Descent (GD):

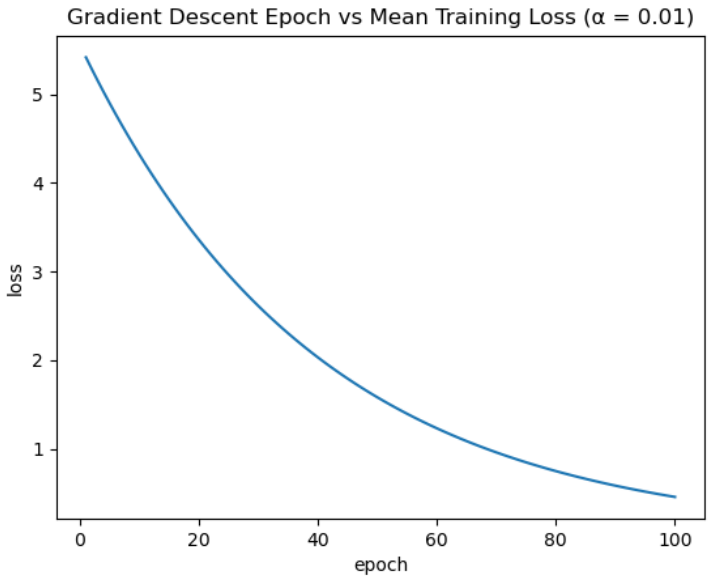

Fig 3: Gradient Descent loss vs epoch with learning rate of 0.01

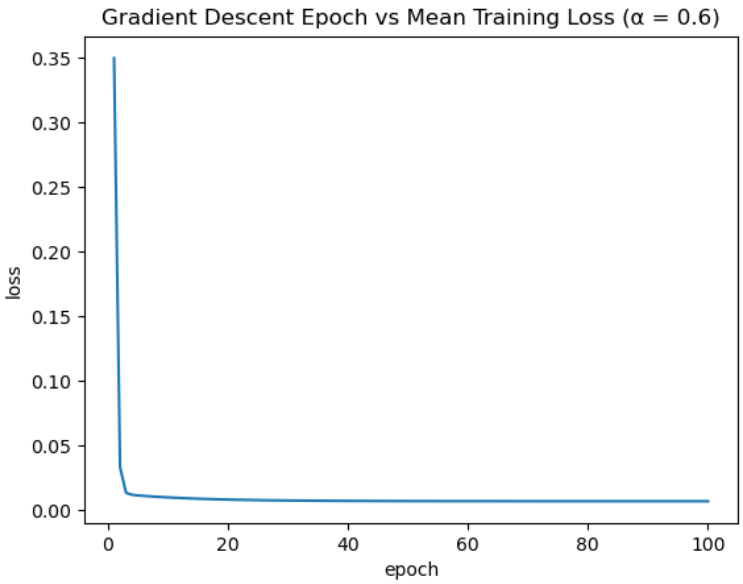

Fig 4: Gradient Descent loss vs epoch with learning rate of 0.6

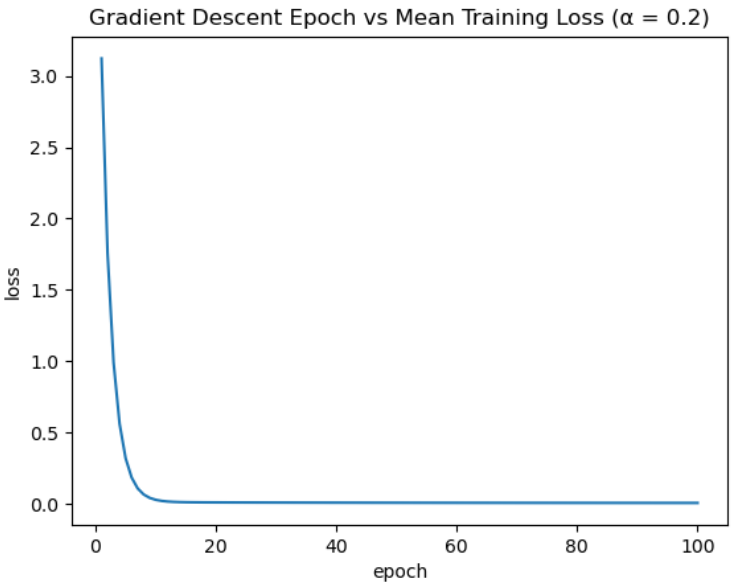

Fig 5: Gradient Descent loss vs epoch with learning rate of 0.2

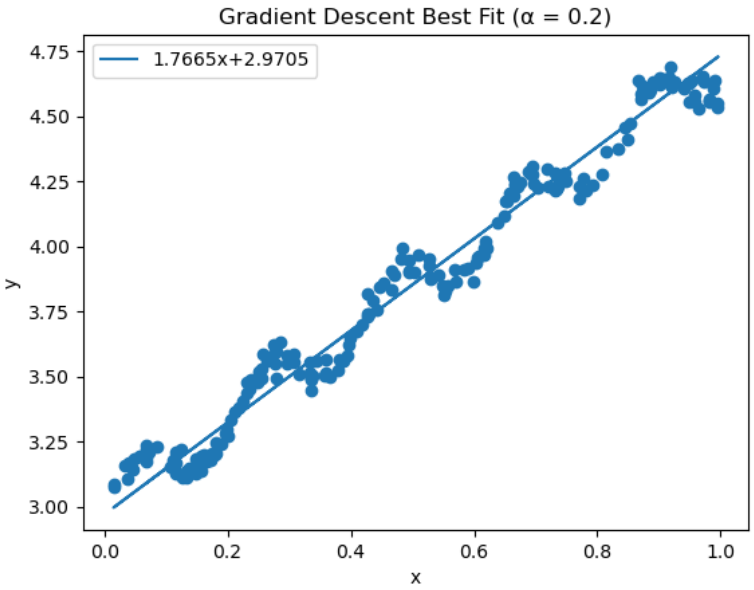

Fig 6: Best-fit LR line with Gradient Descent. Used learning rate = 0.2

C1: The learning rate is a hyperparameter which determines how much to change the model in response to the estimated error. Changing the learning rate changes how quickly the model is adapted to the problem. Regarding the learning rate and Gradient Decent, a small learning rate will cause the line to not nicely fit the data. Additionally, the optimization algorithm that estimates the error gradient won't converge if the learning rate is too low. However, if the learning rate for the Gradient Decent it too large then the model will converge too quickly won't know when to stop. If the learning rate is too large or small, it will lead to suboptimal results. The learning rate needs to be able to show the convergence of the Gradient Decent and produce a well fitted line to the data. An example of a low learning rate can be seen in Figure 3. A learning rate with a too large value can be seen in Figure 4 where the Gradient Decent converged too quickly. Finally, a optimal learning rate for the Gradient Decent can be seen in Figure 5 and 6.

D: Linear Regression with Stochastic Gradient Descent (SGD):

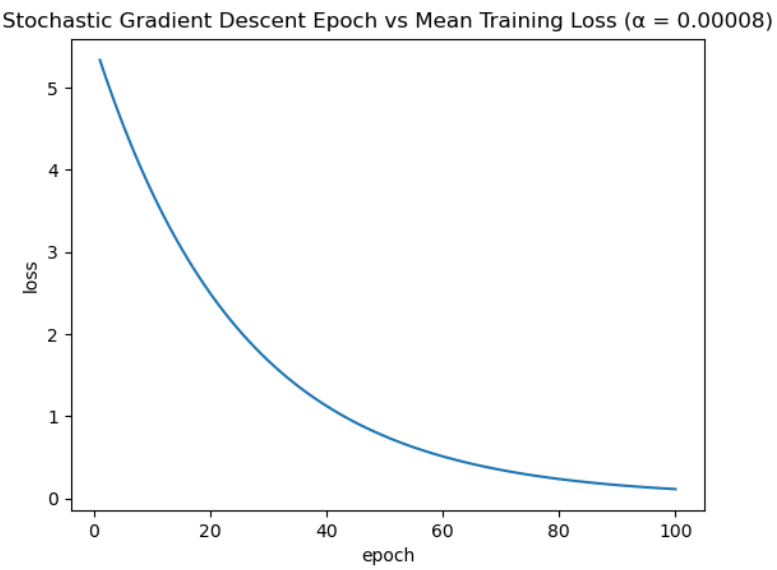

Fig 7: SGD loss vs epoch with learning rate of 0.00008

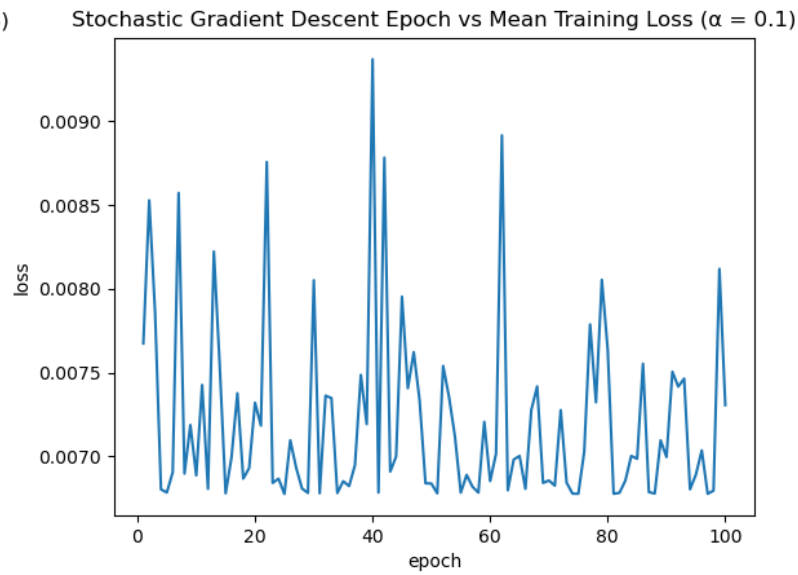

Fig 8: SGD loss vs epoch with learning rate of 0.01

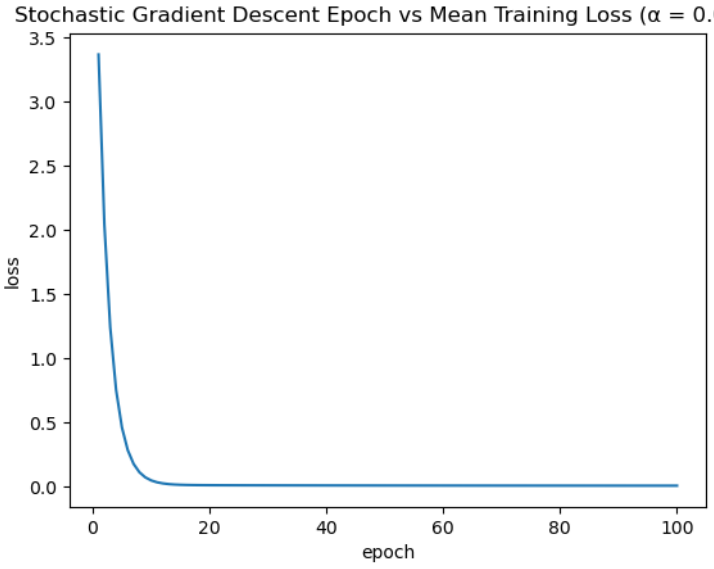

Fig 9: SGD loss vs epoch with learning rate of 0.001

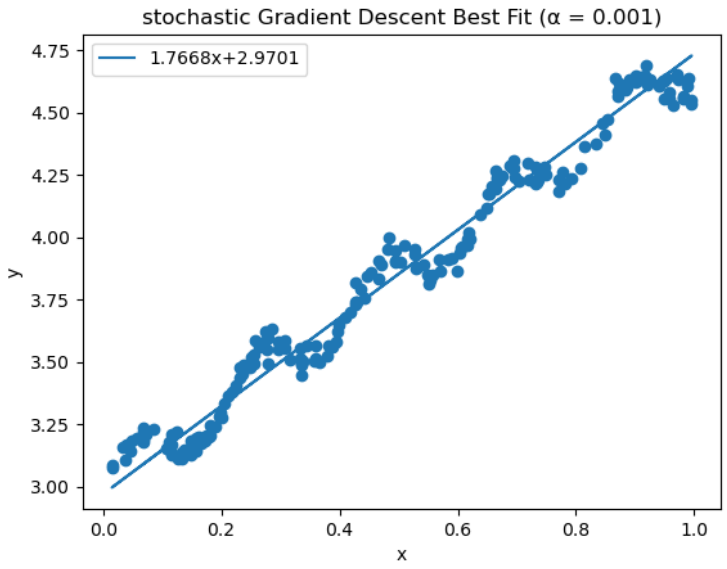

Fig 10: Best-fit LR line with SGD. Used learning rate = 0.001

D1: The effect that learning rate has for a Stochastic Gradient Decent the optimization algorithm is different from a Gradient Decent the optimization algorithm. A Gradient Decent optimization algorithm converges slowly, but a Stochastic Gradient Decent the optimization algorithm converges quickly and doesn't always guarantee convergence. The Stochastic Gradient Decent the optimization algorithm converges is oscillating and can jump to higher contours if the learning rate is too high. The noise of the SGD does help jump out of bad local minima's, therefore the reason why the learning rate must be extremely low for the model not to converge as shown in Fig 7. Consequently, the SGD model has a flaw since if the learning rate is too high, then the model won't converge and will jump to various contours. This is because the optimization algorithm frequently updates the theta. Ultimately, the optimal learning rate for SGD is going to be lower than the optimal learning rate for GD.

E: Linear Regression with Mini-batch Gradient Descent (MiniSGD):

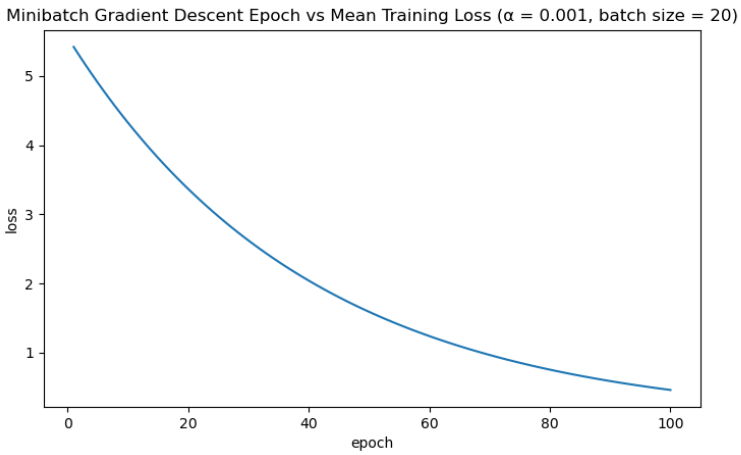

Fig 11: MiniSGD loss vs epoch with learning rate = 0.001 and batch_size = 20

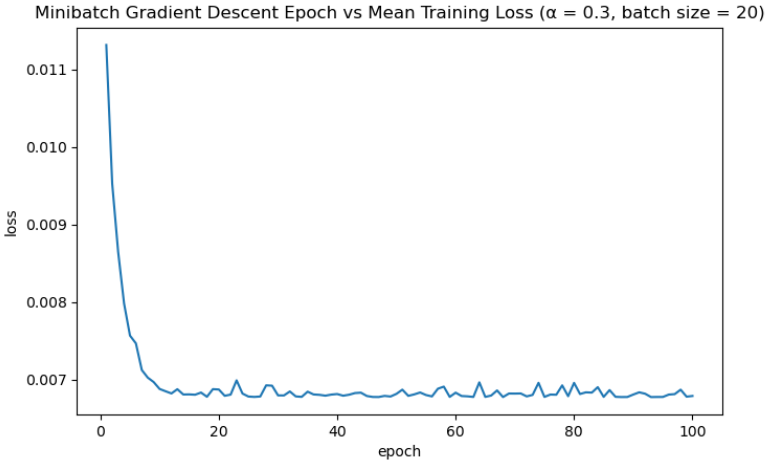

Fig 12: MiniSGD loss vs epoch with learning rate = 0.3 and batch_size = 20

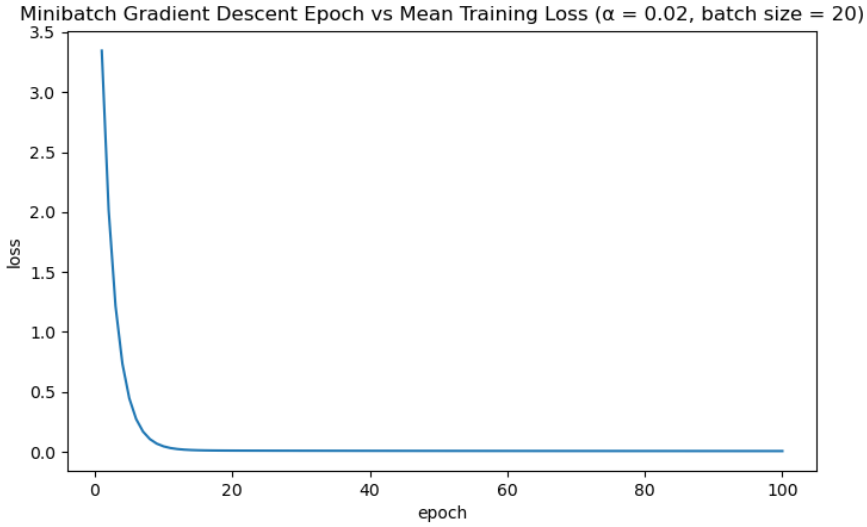

Fig 13: MiniSGD loss vs epoch with learning rate = 0.02 and batch_size = 20

E1: Mini Batch Gradient Descent is a mixture of SGD and Batch Gradient Decent. SGD converges extremely quick, but this also causes a lot of noise. Mini Batch Gradient Descent slows down the computations making it quicker that GD but less noisy than SGD. In Mini Batch Gradient Descent there are two hyperparameters, the learning rate and the batch size. When keeping the batch size constant, we can see how the learning rate affects the Mini Batch Gradient Descent. A too small learning rate will cause the Mini Batch Gradient Descent to not converge. However, this value is a lower value than the GD and a larger value than the SGD. If the learning rate is too large, the convergence will have noise and won't converge. However, the noise is less than the SGD. Finally, for the optimal learning rate the Mini Batch Gradient Descent will converge. Typically, the GD prefers larger learning rates, SGD prefers lower learning rate and Mini Batch Gradient Descent is normally in the middle, but it does also depend on the batch size.

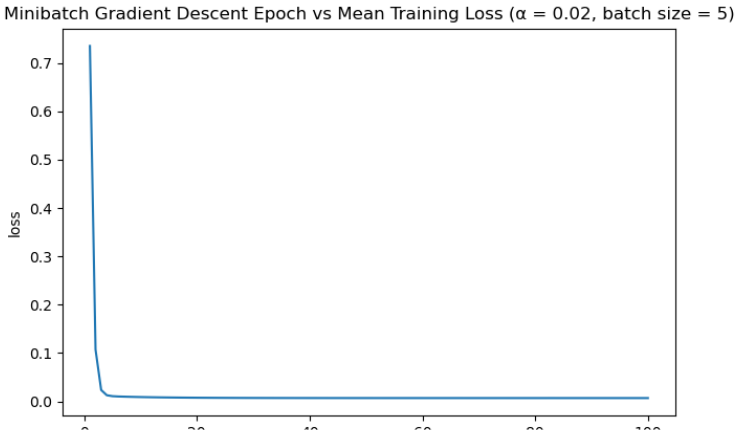

Fig 14: MiniSGD loss vs epoch with learning rate = 0.02 and batch_size = 5

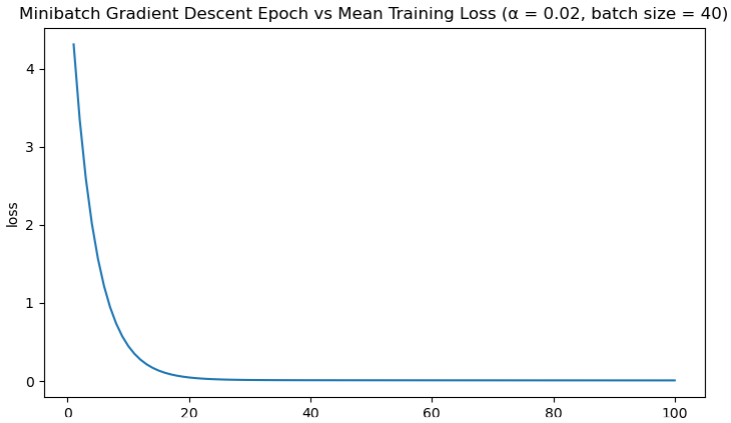

Fig 15: MiniSGD loss vs epoch with learning rate = 0.02 and batch_size = 40

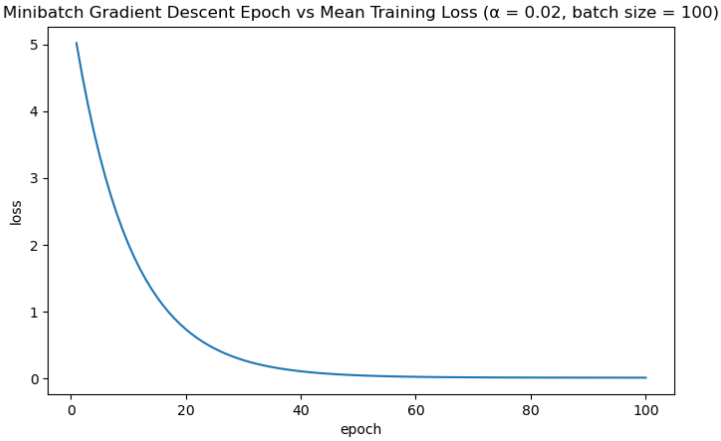

Fig 16: MiniSGD loss vs epoch with learning rate = 0.02 and batch_size = 100

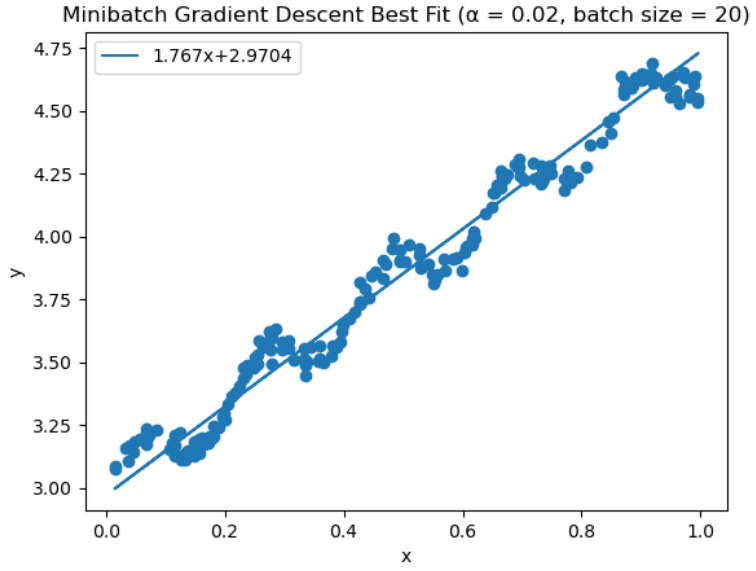

Fig 17: Best-fit LR line with MiniSGD. Here, learning rate = 0.02 and batch_size = 20

E2: The batch size is the major difference between the three types of Gradient Decent. In GD the whole dataset is used per epoch, in the SGD a single example is used at a time in the epoch and in the Mini Batch Gradient Descent a batch of examples are used per epoch. Using the whole dataset per epoch causes the Gradient Decent to converge slowly but using a single example at a time causes more noise. The MiniSGD more stable than the SGD since it is averaging a small number of examples at a time. Using a small batch size causes the MiniSGD to converge quickly as seen in Figure 14. By using a large batch size this is more alike a GD and the convergence is a lot slower. Then an optimal batch size will converge quicker than a GD but slower than a SGD,