Part 1:

Why is gradient descent important in machine learning?

One of the reasons why gradient descent is important as it is used to solve machine learning problems that may occur, it is capable of finding optimal solutions to a large variety of problems. One of its main purposes is to potentially minimize functions but also, use it to update the parameters that our model may have.

How does plain vanilla gradient descent work?

The way in which plain vanilla gradient descent works is by taking small steps into the direction of the minima through taking the gradient of a cost function. In more detail, it is done by calculating an error/loss that is given within each of the samples used in the training set. Once this mode has gone through the network, it updates the model's parameters. One of the importance of this is that it creates more stable gradient descent and can benefit from the vectorisation, helping quicken the process of a models training sample.

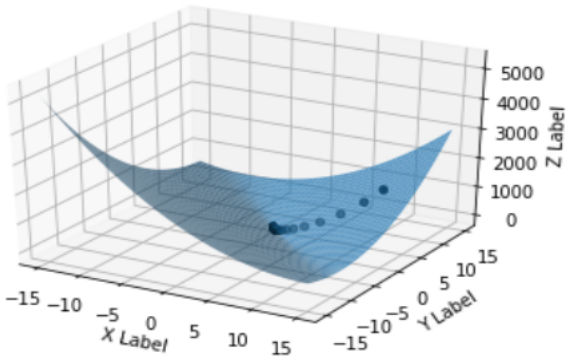
Python Results:

The following plots display how they all converge to a minimum gradient descent. With this, we have managed to find the optimum points within Momentum, NAG and Vanilla. We aimed to produce a 3D plot for the core purpose of being able to produce greater representation power so we can physically examine where the key points are.

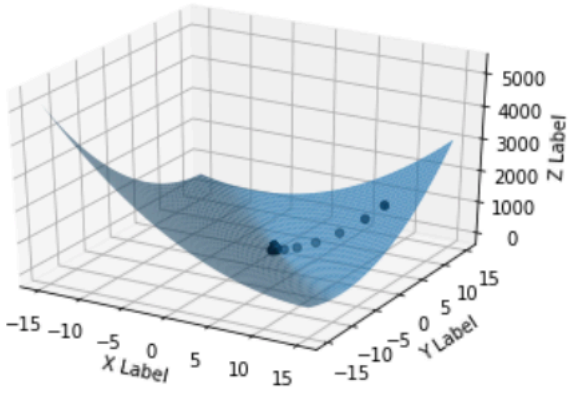
In order to calculate our gradient descent for each of the three we used the formulas and codes introduced under the 'Formulas and calculations' header.

As a result, we can see that the Momentum and NAG plot shows a very clear similarity in comparison to the Vanilla plot. The NAG and Momentum plot displays the clear descent steps taken to reach its lowest point (the area where we aim to reach). With Vanilla gradient we can understand that its descent steps are very close to each other and centralised towards its optimum area. We can clarify that both NAG and Momentum are much clearer representations of our gradient descent as we can clearly visualise it monotonically decreasing.

Momentum:



NAG:



Vanilla: