Question 1: Why is it important to have a learning rate (alpha) that is not too large?

Question 2: How can you do gradient descent when you cannot take the derivative (calculate the gradient) of the function you are trying to minimize?

Question 3: What is the problem with standard gradient descent when using large datasets?

Answer Question 1: Having a learning rate that is too large can cause the update to the weights to be too large so that it overshoots the minimum of the loss function thus diverging. This leads to the problem of preventing gradient descent from finding the minimum of a function. However, you still want the learning rate to not be so small or else the optimization will take a long time to converge. Learning rate is important to balance between not too small and not too large.

Answer Question 2: If the function you are trying to optimize is difficult or impossible to differentiate, you can estimate the gradient using the difference quotient: (f(x+h) - f(x) / h). This is specifically an estimation of the rate of change for a function in small steps “h”, which is exactly what the gradient is representing.

Answer Question 3: In standard gradient descent, the weights are updated from every single example in the dataset. In other words, the gradient must be calculated through the summation of the gradient for every single training example in the dataset. This can get computationally expensive for large datasets. This is why there are different variations of gradient descent such as stochastic and batch gradient descent.