# SDM - Assignment 3, Question 2

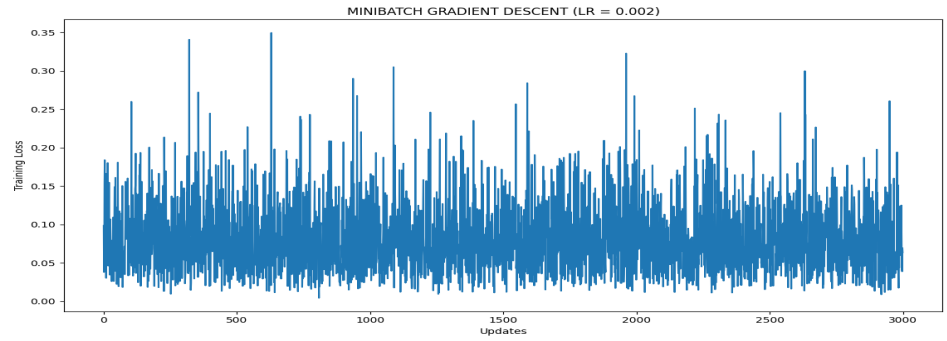
Kousshik Raj (17CS30022)

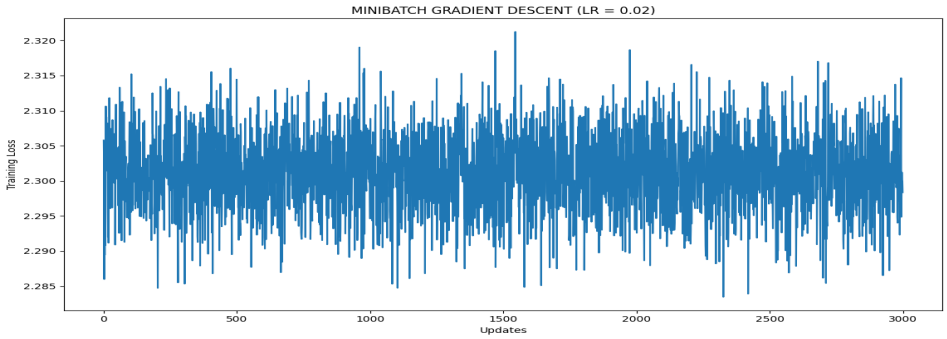
8-11-2020

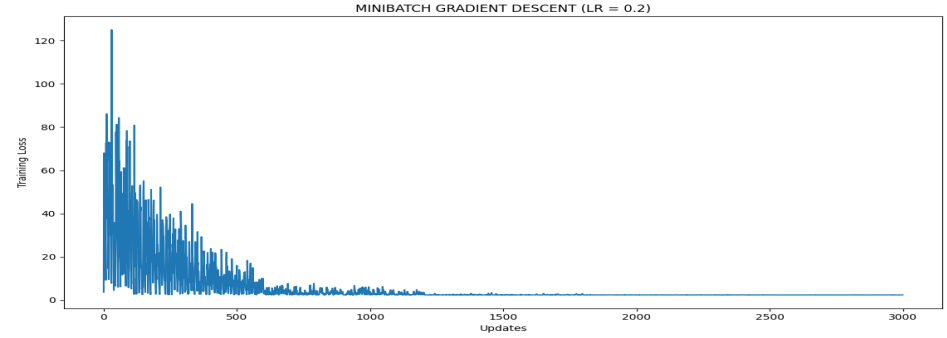
Training a Image Classifier model with convolutional neural network on MNIST handwritten images and comparing the performance of 4 different optimisers (Minibatch Gradient Descent, Momentum Gradient Descent, Nesterov Gradient Descent, Adam) for multiple learning rates.

## 1 Training Loss After Every Update

### 1.1 Minibatch Gradient Descent

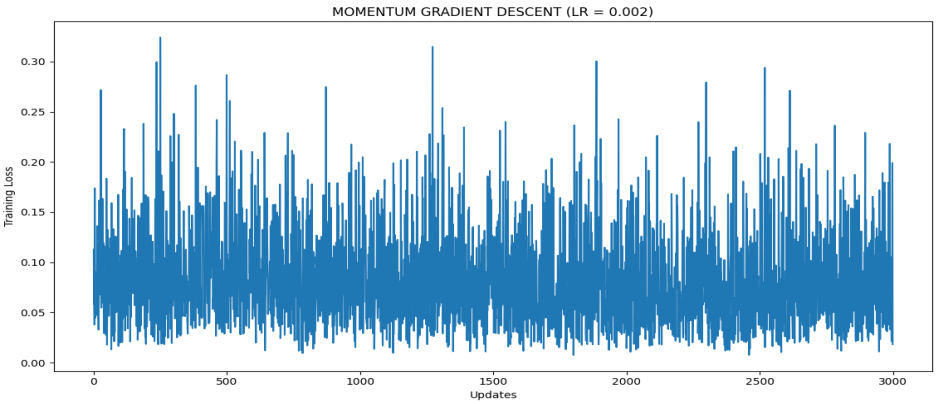


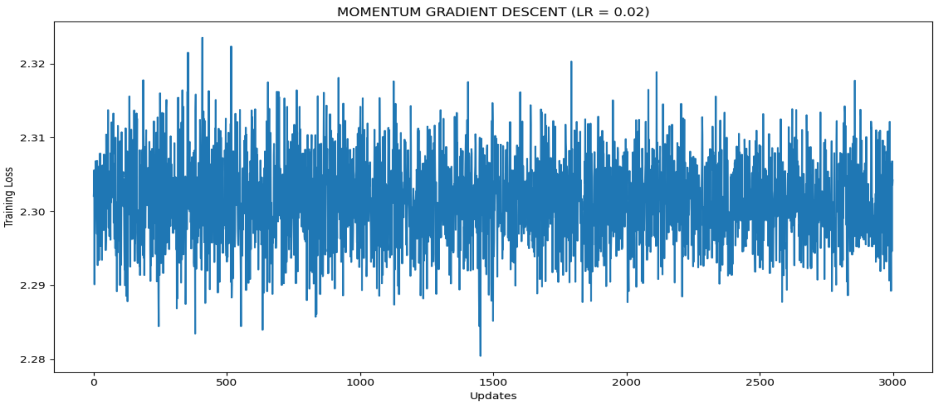



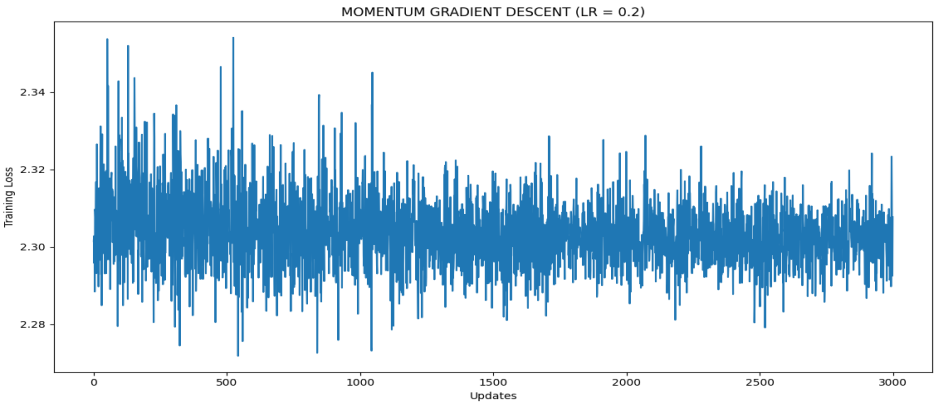




Since the loss is calculated for different mini batches after every update, the loss fluctuates quite a bit, but as can be seen from the plots, the model with learning rate 0.002 oscillates around a significantly lower value than the other models. This might be because with a high learning rate, the model cannot easily converge and thus the training loss usually increases in such cases. Especially, the last plot converges on a local minima, and hence has a worse performance.

## 1.2 Momentum Gradient Descent

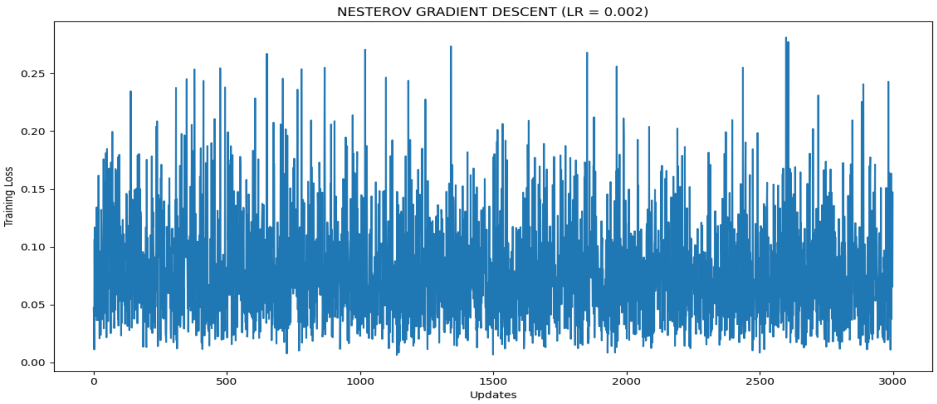


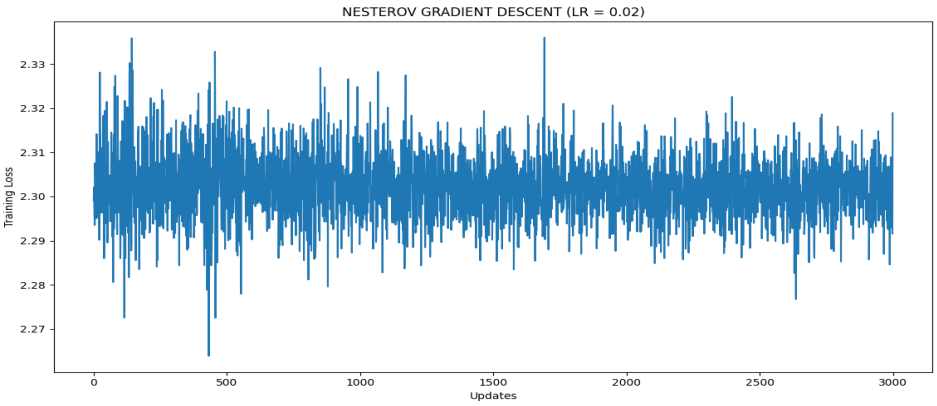



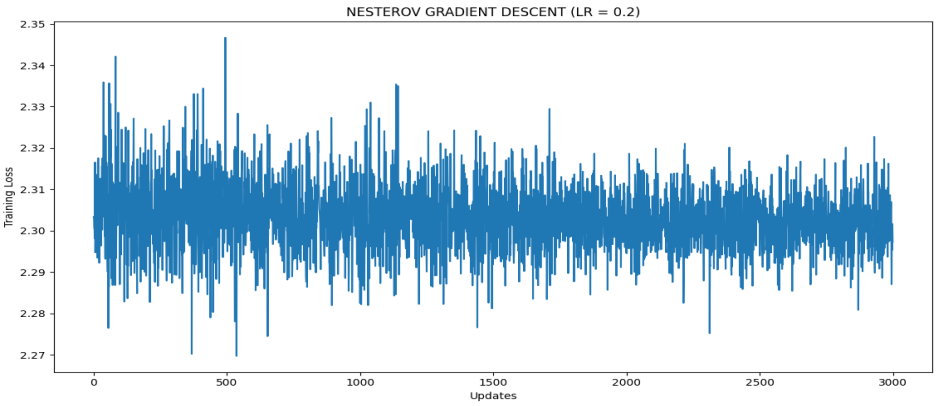




One observation that is common to all the three plots is the high fluctuations in the training loss. One aspect of the reason, is that the loss is calculated for different mini batches, but the main reason is the inherent characteristic of momentum GD, which tends to retain the direction of previous updates. Initially, there are quite large oscillations, but as the number of epochs increase, the magnitude of oscillations decreases which is quite obvious in all the plots. But, the last 2 models converge at a local minima that is significantly worse than the first model, which is because of the high learning rate. High learning rate tends to bring about a drastic change in the parameters which in most case results in a increase in the training loss.

## 1.3 Nesterov Gradient Descent

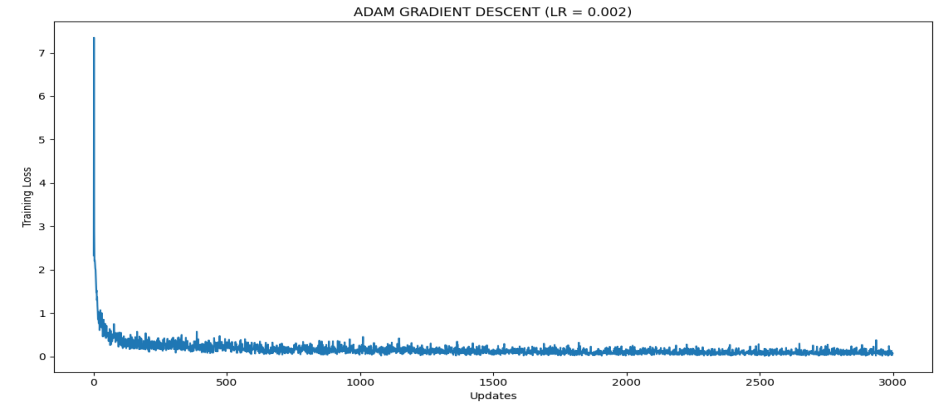


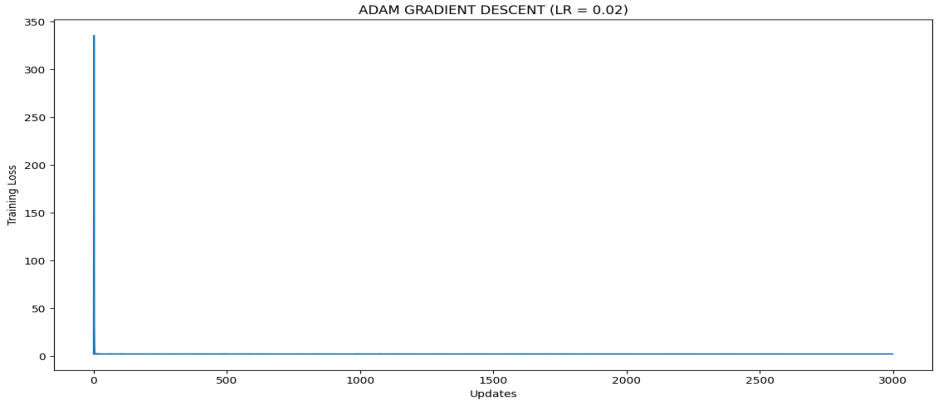



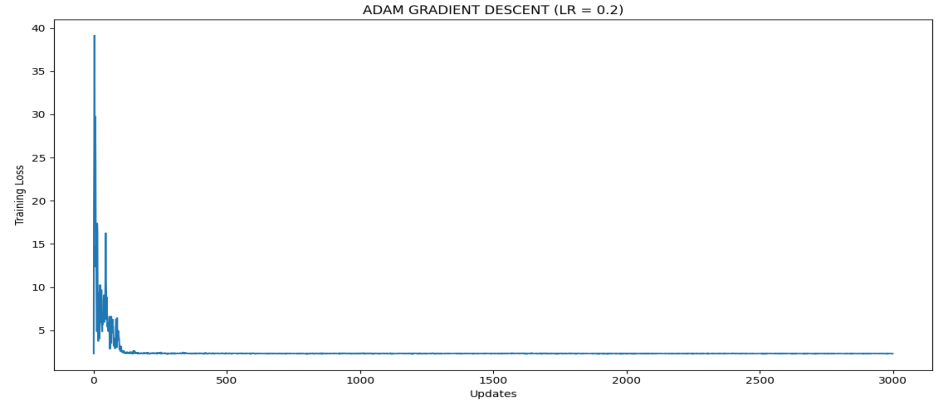




The Nesterov GD is quite similar to momentum GD, but differs in the way the gradients are calculated. Nesterov GD calculates gradients of parameters after moving in the direction of previous updates, which is not the case in the momentum GD. This change of the calculation of gradients resolves the problem of large oscillations in momentum GD. This is especially obvious in cases of high learning rate, as can be seen from the last 2 plots. But, the problem of high learning rate resulting in high training error is still the same as the previous optimisers. The model with learning rate 0.002 converges around 0.08 loss, whereas the others converge on a loss of 2.30.

## 1.4 Adam Optimiser

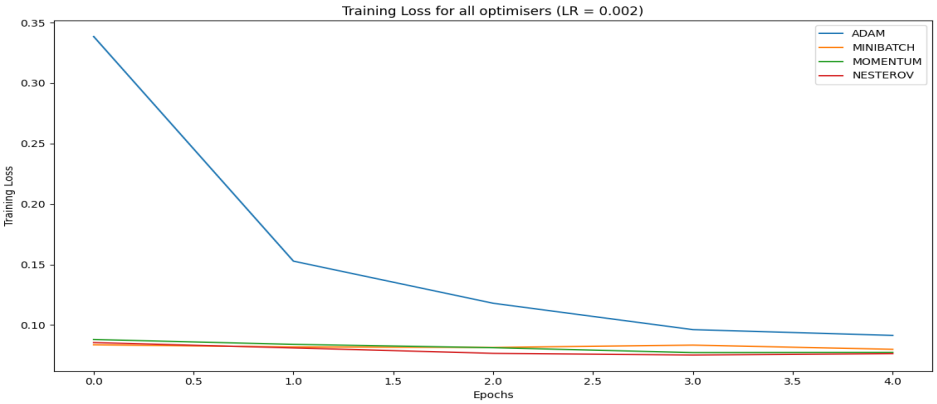


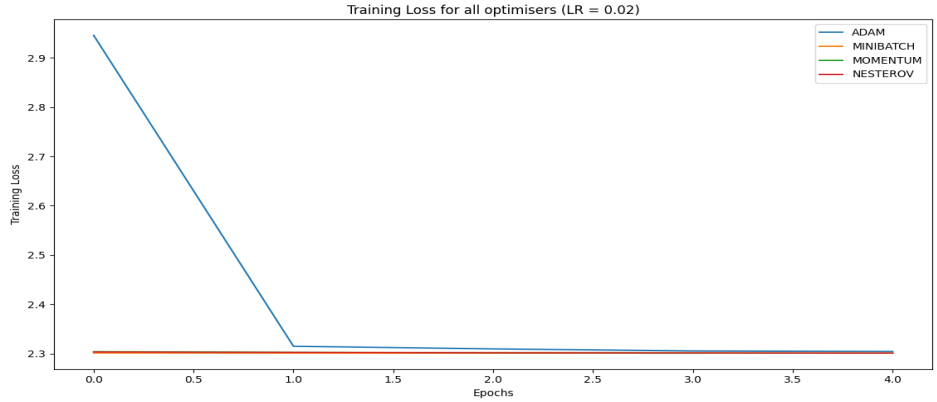



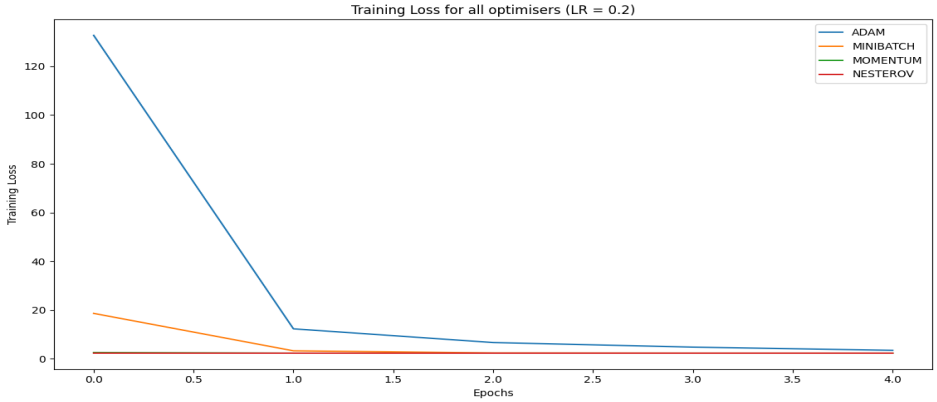




In these plots, we can see that all the different learning rates have a similar convergence pattern, except for the fact that higher learning rates converge on a local minima instead of a global one, which results in a drastically worse performance than the one with a low learning rate. The later 2 models converge on a loss of 2.3 whereas the first one converges on loss of 0.1, similar to the all the other optimisers. The only thing that might be different from the other optimisers is the lower fluctuations and the slower convergence due to the learning rate decay.

## 2 Training Loss After Every Epoch

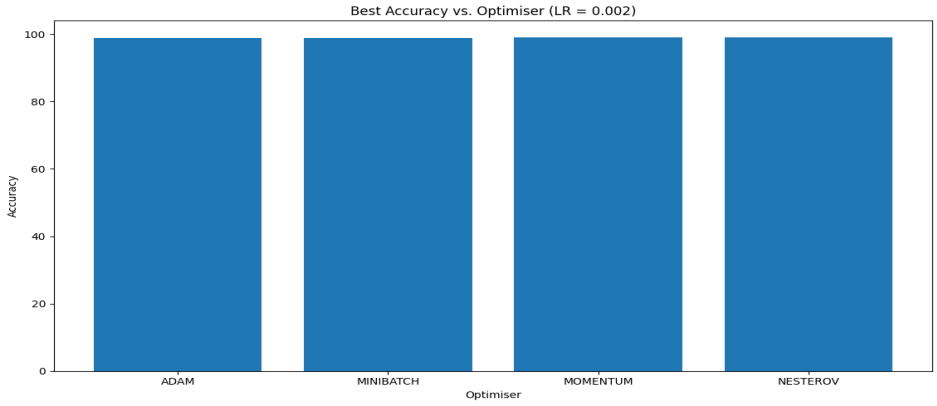


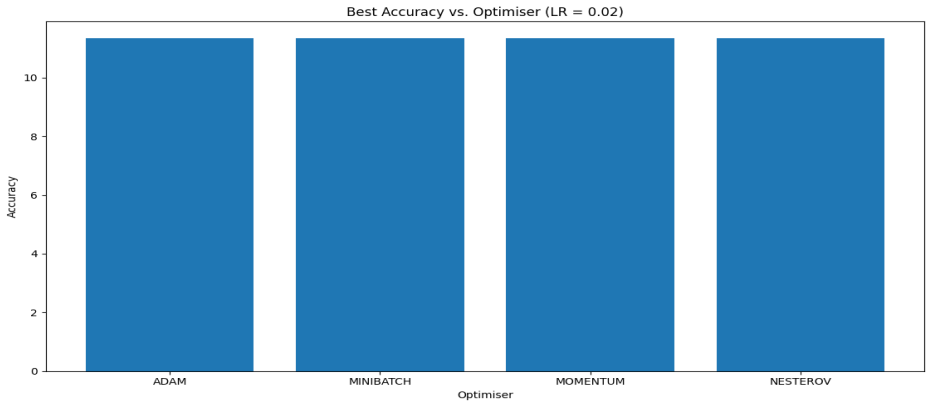



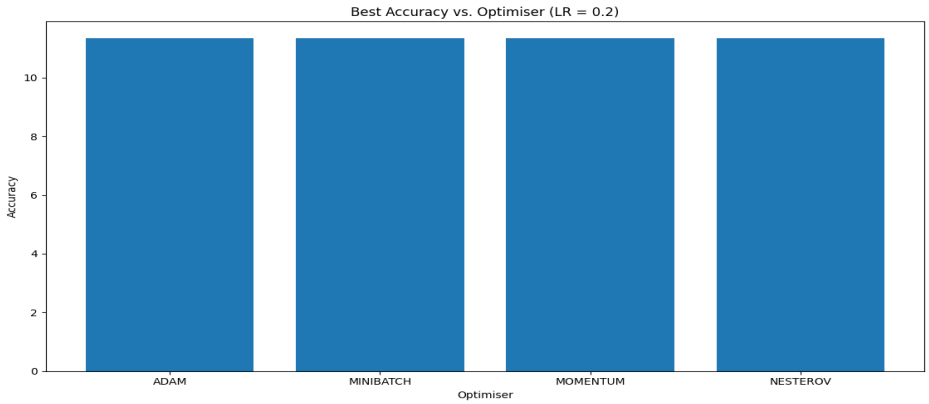




One fact that is common in all these plots is the slow convergence of Adam optimiser, irrespective of whether it converges to a lower or a higher value, because of the high decay of learning rate. But this might be helpful in other cases, where a low learning rate is required toward the end for a proper convergence. The difference between other optimisers is not significant when they have the same starting point, especially at higher rates, as at a higher rate they all converge on the same undesirable local minima. The previous plots are also consistent with these graphs.

# 3 Test Accuracy




As was earlier seen, the models with high learning rate have a significantly worse performance. Models with learning rate 0.002, irrespective of the optimiser used, achieved an accuracy of nearly 99% in the test set, whereas all the models with higher learning rate have only an accuracy of around 11%, which is not much better than making an uniformly random guess. Hence, from this we can see the importance of learning rate, and how it influences the model convergence.