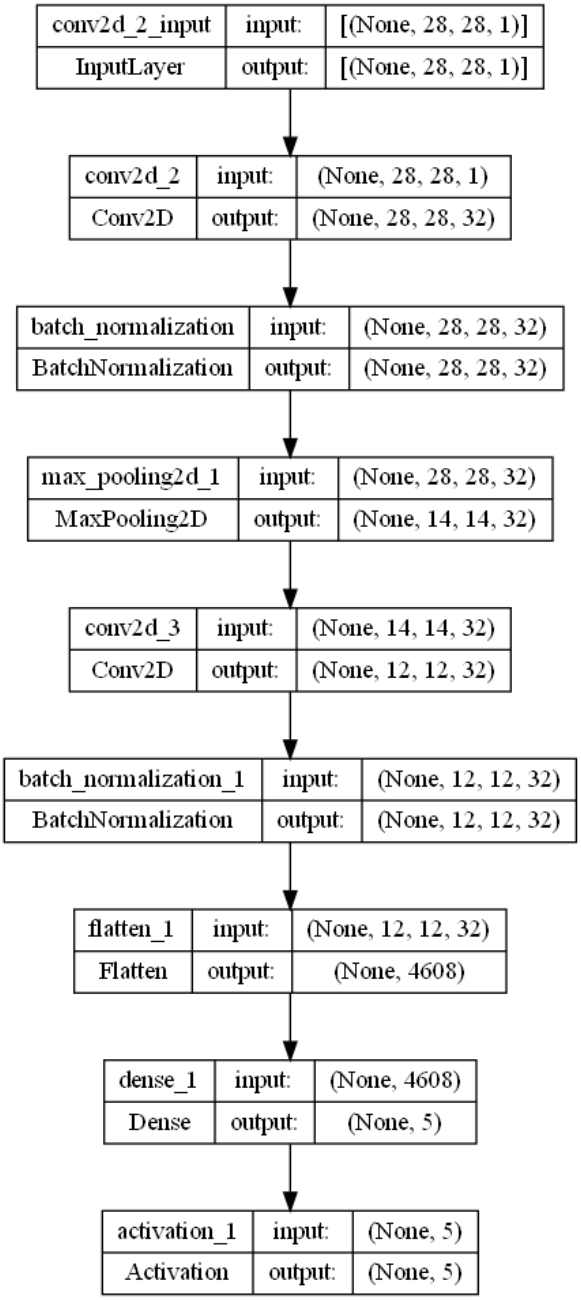
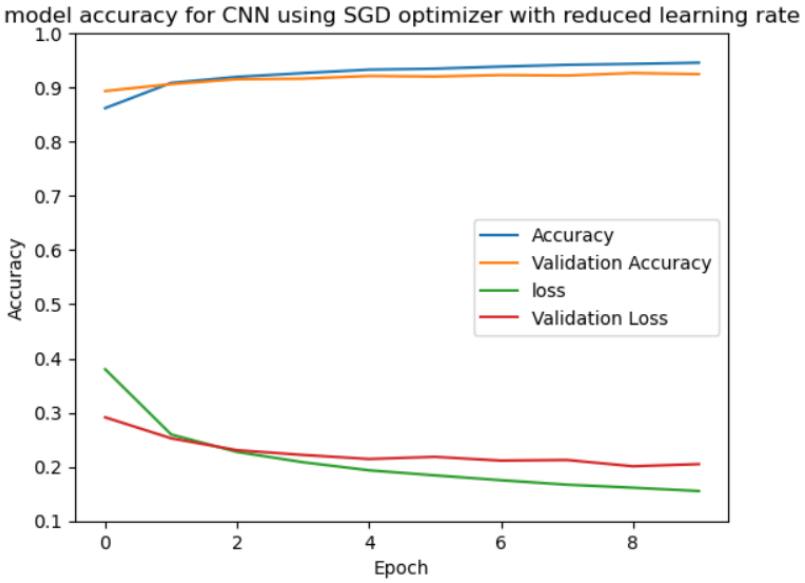
Question2: modifying the model.

To do this we have considered few models. Out first approach is to not to change the model, instead modify the hyperparameters such as learning rate. Reducing the learning rate from 0.01 to 0.001 which resulted in higher test accuracy. Reducing LR increased the computation.

`Model 1: Using Learning Rate: 0.001, Epoch: 10, Test accuracy: 92.40000247955322`





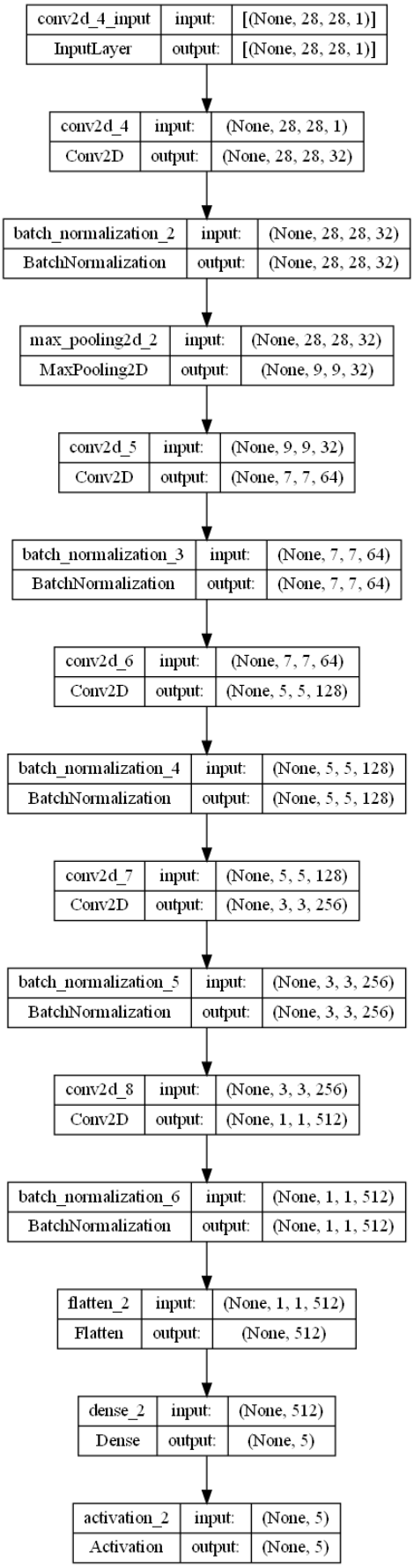
The above model is a neural network that has been trained for 10 epochs on a certain dataset, with a learning rate of 0.001. Based on the output, the model seems to be performing fairly well on the given task, achieving an accuracy of 92.47% on the validation set and 91.87% on the test set.

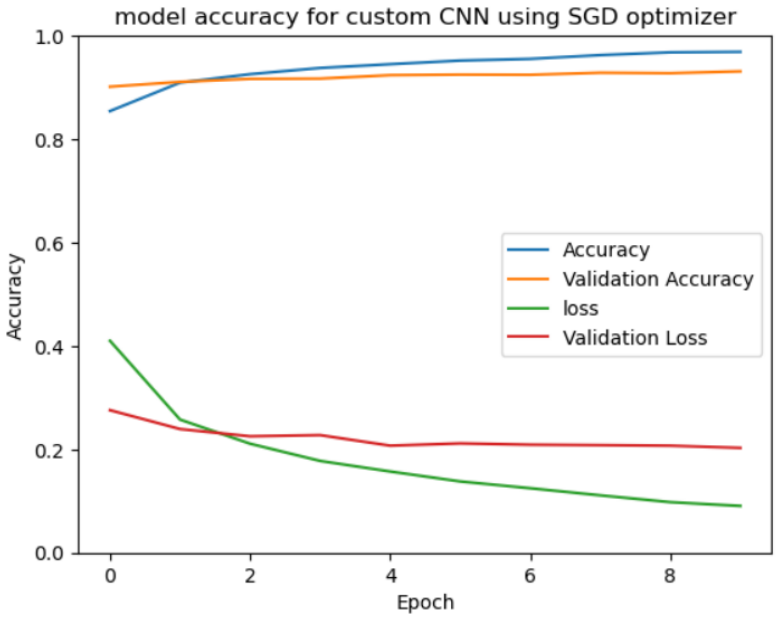
During training, the model was able to steadily improve its accuracy on the validation set from 89.35% in the first epoch to 92.47% in the tenth epoch. This suggests that the model was able to effectively learn from the training data and generalize well to unseen data.

```
Epoch 1/10
2700/2700 [==============================] - 8s 3ms/step - loss: 0.3800 - accuracy: 0.8634 - val_loss: 0.2903 - val_accuracy:
0.8950 - lr: 0.0010
Epoch 2/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.2643 - accuracy: 0.9072 - val_loss: 0.2620 - val_accuracy:
0.9048 - lr: 0.0010
Epoch 3/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.2341 - accuracy: 0.9193 - val_loss: 0.2398 - val_accuracy:
0.9147 - lr: 0.0010
Epoch 4/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.2162 - accuracy: 0.9247 - val_loss: 0.2246 - val_accuracy:
0.9198 - lr: 0.0010
Epoch 5/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.2031 - accuracy: 0.9285 - val_loss: 0.2154 - val_accuracy:
0.9205 - lr: 0.0010
Epoch 6/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.1929 - accuracy: 0.9318 - val_loss: 0.2243 - val_accuracy:
0.9198 - lr: 0.0010
Epoch 7/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.1847 - accuracy: 0.9358 - val_loss: 0.2103 - val_accuracy:
0.9222 - lr: 0.0010
Epoch 8/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.1763 - accuracy: 0.9381 - val_loss: 0.2132 - val_accuracy:
0.9243 - lr: 0.0010
Epoch 9/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.1697 - accuracy: 0.9402 - val_loss: 0.2043 - val_accuracy:
0.9222 - lr: 0.0010
Epoch 10/10
2700/2700 [==============================] - 7s 3ms/step - loss: 0.1646 - accuracy: 0.9430 - val_loss: 0.1973 - val_accuracy:
0.9282 - lr: 0.0010
313/313 [==============================] - 1s 2ms/step - loss: 0.2197 - accuracy: 0.9240
Test accuracy: 92.40000247955322
```

Modified Model 2: `Test accuracy: 92.75000095367432`

Learning Rate: 0.001, Epoch:10, Optimizer: SGD, Loss: Categorical cross entropy





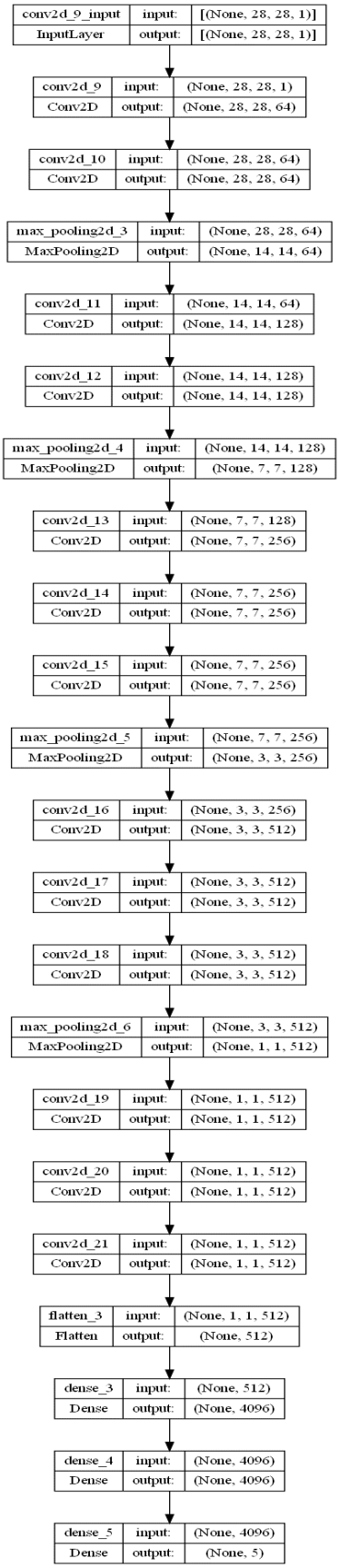
Based on the given output, the model seems to perform well with an increasing accuracy over the training epochs. The training accuracy is at 96.90% while the validation accuracy is at 93.12%, indicating that the model generalizes well to unseen data.

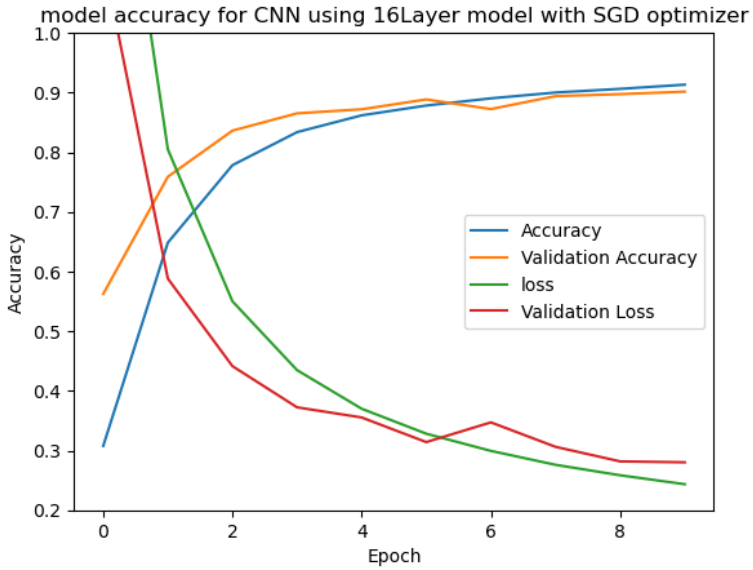
The model also seems to avoid overfitting, as the validation loss remains relatively consistent throughout the training process. The test accuracy of 92.75% is also quite good, indicating that the model performs well on new and unseen data.

Overall, this model seems to be performing well and could be considered a good model for the given task.

```
Epoch 1/10
2700/2700 [==============================] - 14s 5ms/step - loss: 0.4099 - accuracy: 0.8544 - val_loss: 0.2758 - val_accuracy:
0.9015 - lr: 0.0010
Epoch 2/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.2576 - accuracy: 0.9094 - val_loss: 0.2393 - val_accuracy:
0.9108 - lr: 0.0010
Epoch 3/10
2700/2700 [==============================] - 12s 5ms/step - loss: 0.2110 - accuracy: 0.9257 - val_loss: 0.2253 - val_accuracy:
0.9167 - lr: 0.0010
Epoch 4/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.1776 - accuracy: 0.9377 - val_loss: 0.2275 - val_accuracy:
0.9172 - lr: 0.0010
Epoch 5/10
2700/2700 [==============================] - 12s 5ms/step - loss: 0.1572 - accuracy: 0.9450 - val_loss: 0.2072 - val_accuracy:
0.9238 - lr: 0.0010
Epoch 6/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.1380 - accuracy: 0.9520 - val_loss: 0.2115 - val_accuracy:
0.9248 - lr: 0.0010
Epoch 7/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.1248 - accuracy: 0.9553 - val_loss: 0.2090 - val_accuracy:
0.9245 - lr: 0.0010
Epoch 8/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.1111 - accuracy: 0.9626 - val_loss: 0.2082 - val_accuracy:
0.9285 - lr: 0.0010
Epoch 9/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.0979 - accuracy: 0.9680 - val_loss: 0.2071 - val_accuracy:
0.9277 - lr: 0.0010
Epoch 10/10
2700/2700 [==============================] - 13s 5ms/step - loss: 0.0906 - accuracy: 0.9690 - val_loss: 0.2029 - val_accuracy:
0.9312 - lr: 0.0010
313/313 [==============================] - 2s 4ms/step - loss: 0.2221 - accuracy: 0.9275
Test accuracy: 92.75000095367432
```

Model3: Using 16 layer CNN : Test accuracy: 89.48000073432922





The two models (Model2 and Model3) provided have significantly different architectures and training histories.

Model 2 is a CNN model that has 10 epochs of training with a decreasing learning rate. It achieves a test accuracy of 92.75%. Model 3 is also a CNN model with 10 epochs of training but with a different architecture, achieving a test accuracy of 89.11%.

In comparing the two models, we can see that Model 2 has a higher test accuracy than Model 3. It is also noteworthy that the validation accuracy of Model 2 is consistently higher than that of Model 3, indicating that it is more effective in preventing overfitting.

Additionally, the training times for each epoch of Model 2 is much faster than that of Model 3, suggesting that it may be a more efficient model to use for larger datasets.

Overall, we can conclude that Model 2 is the better model of the two due to its higher accuracy, faster training time, and better prevention of overfitting.

```
Epoch 1/10
2700/2700 [==============================] - 64s 23ms/step - loss: 1.5375 - accuracy: 0.3078 - val_loss: 1.1224 - val_accuracy:
0.5625 - lr: 0.0010
Epoch 2/10
2700/2700 [==============================] - 62s 23ms/step - loss: 0.8046 - accuracy: 0.6485 - val_loss: 0.5882 - val_accuracy:
0.7588 - lr: 0.0010
Epoch 3/10
2700/2700 [==============================] - 62s 23ms/step - loss: 0.5501 - accuracy: 0.7783 - val_loss: 0.4414 - val_accuracy:
0.8362 - lr: 0.0010
Epoch 4/10
2700/2700 [==============================] - 98s 36ms/step - loss: 0.4347 - accuracy: 0.8337 - val_loss: 0.3724 - val_accuracy:
0.8652 - lr: 0.0010
Epoch 5/10
2700/2700 [==============================] - 98s 36ms/step - loss: 0.3700 - accuracy: 0.8618 - val_loss: 0.3554 - val_accuracy:
0.8720 - lr: 0.0010
Epoch 6/10
2700/2700 [==============================] - 97s 36ms/step - loss: 0.3282 - accuracy: 0.8783 - val_loss: 0.3141 - val_accuracy:
0.8885 - lr: 0.0010
Epoch 7/10
2700/2700 [==============================] - 97s 36ms/step - loss: 0.2994 - accuracy: 0.8903 - val_loss: 0.3472 - val_accuracy:
0.8723 - lr: 0.0010
Epoch 8/10
2700/2700 [==============================] - 7351s 3s/step - loss: 0.2761 - accuracy: 0.9001 - val_loss: 0.3062 - val_accuracy:
0.8940 - lr: 0.0010
Epoch 9/10
2700/2700 [==============================] - 64s 24ms/step - loss: 0.2586 - accuracy: 0.9062 - val_loss: 0.2818 - val_accuracy:
0.8973 - lr: 0.0010
Epoch 10/10
2700/2700 [==============================] - 65s 24ms/step - loss: 0.2436 - accuracy: 0.9132 - val_loss: 0.2803 - val_accuracy:
0.9015 - lr: 0.0010
313/313 [==============================] - 4s 11ms/step - loss: 0.2795 - accuracy: 0.8948
Test accuracy: 89.48000073432922
```