As default hyperparameters, I had:

* 2 layers
* A 32x32 embedding layer
* A batch size of 64
* A learning rate of 0.0001
* 30 epochs

This allowed my autoencoder model to achieve a training accuracy of 0.641 and a validation accuracy of 0.635.

------------------------------------Test 1------------------------------------

The first hyperparameter I changed was the dimension of the embedding layer, decreasing it from 32x32 to 16x16. I did this because I thought it would force the autoencoder to be more specific in which values it chose to be important, and to keep in the low-dimensional embedding. The ultimate goal is to train the autoencoder to recognize which feature values are important in making an accurate prediction. I kept all other hyperparameter values the same. With the first set of values, I achieved a training accuracy of 0.645 and a validation accuracy of 0.639.

------------------------------------Test 2------------------------------------

The results of Test 1 demonstrated that the model experienced more overfitting compared to the default model, since the training accuracy is higher but the validation accuracy is lower. To reduce the likelihood of overfitting, I decided to increase the learning rate from 0.0001 to 0.001, while keeping the dimensions of the embedding layer as 32x32. This second set of values produced a training accuracy of 0.654 and a validation accuracy of 0.651.

------------------------------------Test 3------------------------------------

The accuracy values of Test 2 are both higher than the default set’s accuracy. Furthermore, the loss values generated by Test 2 are both lower than in the default set. Overall, this shows that the hyperparameter values chosen in Test 2 were a good fit to the data. In Test 3, I decided to decrease the batch size from 64 to 32, while keeping all other hyperparameter values the same as in Test 2. This produced a training accuracy of 0.661 and a validation accuracy of 0.657.

------------------------------------Test 4------------------------------------

As a last hyperparameter test, I wanted to see whether the high accuracy achieved in the last test was more influenced by the batch size or the dimension of the encoding layer. From Test 2, I learned that a learning rate of 0.001 produced higher accuracies than a learning rate of 0.0001 when the batch size was 64. From Test 3, I learned that lowering the batch size from 64 to 32 produced higher accuracies than in Test 2. However, the difference between the training and validation loss was greater in Test 3 than Test 2, indicating more overfitting in Test 3 than in Test 2. I wanted to reduce this overfitting, and Test 2 indicated that setting the encoding dimension to 32 seems to reduce the likelihood of overfitting. The hyperparameter values I used in Test 4 are the same as the values in Test 2, except the dimension of the embedding layer was increased to 32x32. This produced a training accuracy of 0.668 and a validation accuracy of 0.661.

As a side note, I also tried adding more fully-connected layers by adding a hidden layer with 40 units, connected to the input layer through a ReLU activation function, but this produced a training accuracy of 0.657 and a validation accuracy of 0.630. Furthermore, the training loss was 0.006 and the validation loss was 0.015, which indicated overfitting compared to other test models.