Default

* Number of hidden units: 32
* Batch size: 32
* Learning rate: 1e-4
* Number of epochs: 30
* Type of output pooling: default
* Number of FC layers: 1

Default model produced a training accuracy of 0.885 and a validation accuracy of 0.873.

Model 1

* Number of hidden units: 64
* Batch size: 32
* Learning rate: 1e-4
* Number of epochs: 30
* Type of output pooling: default
* Number of FC layers: 1

To increase the training and validation accuracy, I decided to increase the number of hidden units used in the fully-connected layer. This produced a training accuracy of 0.880 and a validation accuracy of 0.891. Model 1 produced a higher validation accuracy then the default model. However, both the training and validation loss of Model 1 were higher than the default model.

Model 2

* Number of hidden units: 64
* Batch size: 32
* Learning rate: 1e-4
* Number of epochs: 30
* Type of output pooling: max pooling
* Number of FC layers: 1

To minimize the training and validation loss, I decided to use max pooling over the entire array in the output layer as was suggested in the lab handout. This allowed me to achieve a training accuracy of 0.981 and a validation accuracy of 0.964. Furthermore, this decreased both the training and validation loss to 0.037 and 0.018 respectively.

Model 3

* Number of hidden units: 64
* Batch size: 32
* Learning rate: 5e-4
* Number of epochs: 30
* Type of output pooling: max pooling
* Number of FC layers: 1

I chose to increase the learning rate in model 3 because I wanted to achieve a smaller loss. Model 3 produced a training accuracy of 0.9997 and a validation accuracy of 0.986, and a training loss of 0.0038 and a validation loss of 0.0021. The loss of model 3 is 10x smaller than the loss for model 2, so choosing a larger learning rate allowed model 3 to reach a better local/global minimum than model 2.

Model 4

* Number of hidden units in layer 1: 64
* Number of hidden units in layer 2: 32
* Activation function: ReLU
* Batch size: 32
* Learning rate: 5e-4
* Number of epochs: 30
* Type of output pooling: max pooling
* Number of FC layers: 2

I chose to increase the number of fully-connected layers in model 4 to see if I could increase the validation accuracy any further. I kept the number of hidden units in layer 1 the same as the previous 3 models, and used 32 hidden units in the second fully-connected layer. I chose to use the ReLU activation function between the 2 hidden layers because it doesn’t have the vanishing gradient problem as in the sigmoid activation function, and it converges faster than the tanh activation function. The training accuracy of model 4 is 0.896, the validation accuracy of model 4 is 0.966, the training loss is 0.023, and the validation loss is 0.042. This is a worse set of results than model 3, which suggests that adding more layers makes training the model more difficult.

From the 4 models and the default model, I chose to use the hyperparameters of model 3 because it produced the highest accuracies and the lowest losses.