1. Implementing backpropagation

This homework builds on your code from the previous homework to complete your neural network implementation. To do that, you will need to implement the backward_pass function. You should leverage the cached intermediate steps and output in your gradient computations. Note that it is possible to simplify your computations by using only matrix operations rather than iteration.

2. Evaluating Learning Performance

Make plots of the loss over time and the accuracy of the performance over time while varying the following parameters.

- (1) Learning rate range: explore 3 different rates between 0.0001 and 1.0 and see what happens to the loss over time. Note that it is better to set the learning rate too small than too large. If you are seeing losses of Infinity or NaN, your learning rate is likely too high.
- (2) Hidden layer size: explore 3 sizes between 5 and 40 nodes. Increasing the hidden layer will improve how well the neural network is able to fit the training data, but can lead to over-fitting. More hidden nodes will increase training time and may also make it harder to train the neural network since there are more parameters that need to be learned.