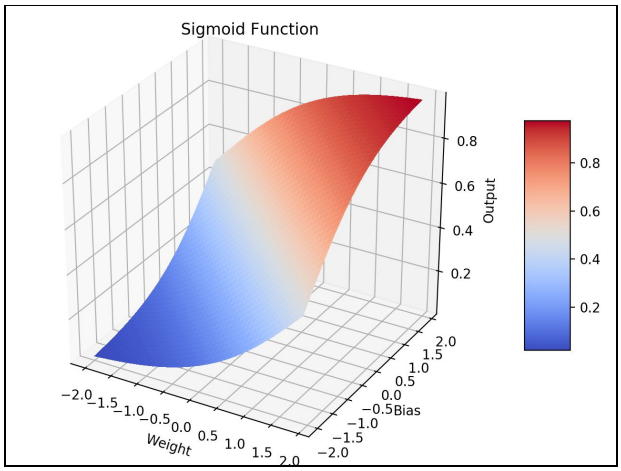
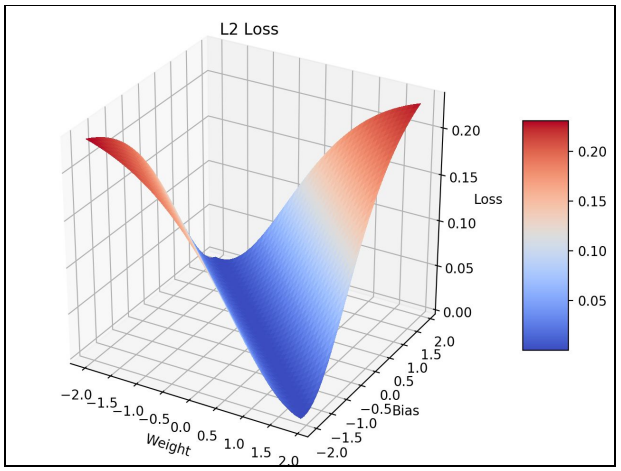
Luigi Mangione
Collaborators: Brian Chirikjian

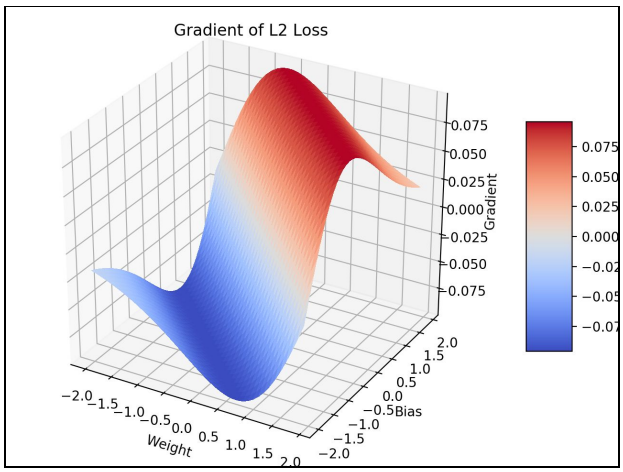
12/11/19

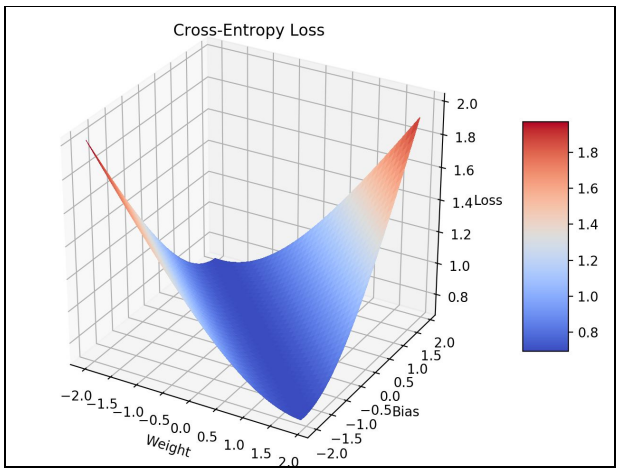
CIS 581: Project 4 - Deep Learning Basics

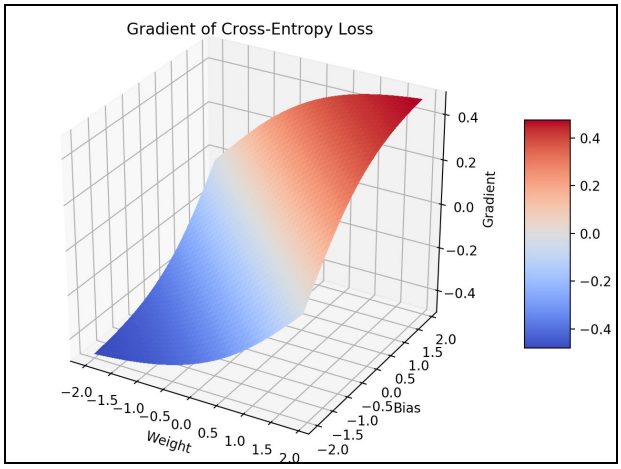
1 Plot Loss and Gradient









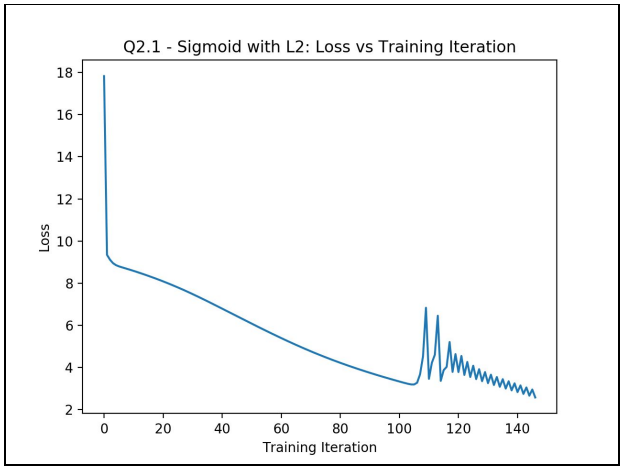


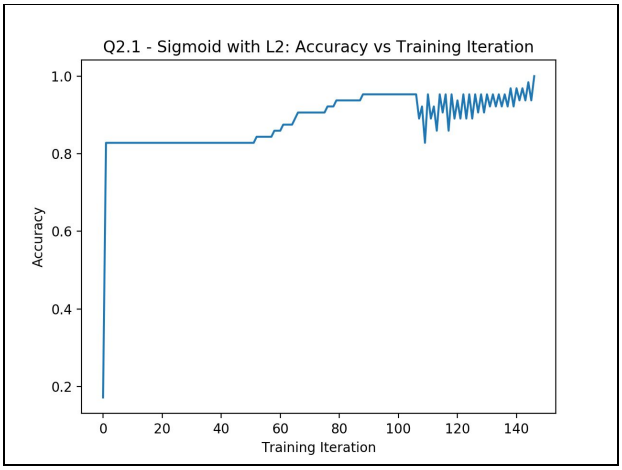
L2 loss represents the squared difference between the predicted and actual value for a given example. Cross-entropy loss represents the negative log of the predicted label, scaled by how far off it is from the true label.

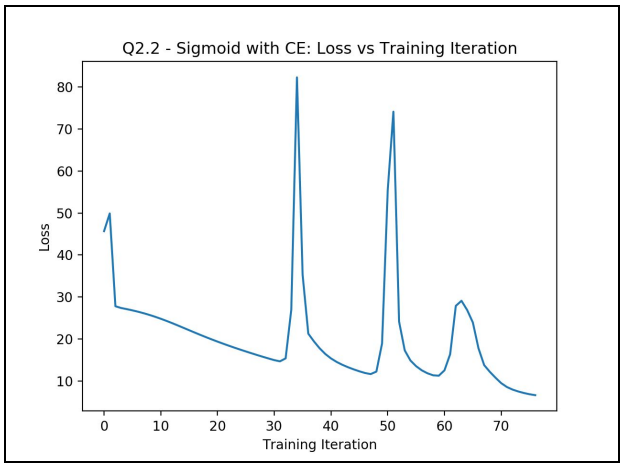
Comparing the gradients of both L2 and cross entropy-loss, we see that the gradients for cross-entropy loss have a significantly higher magnitude. Thus, gradient descent using cross-entropy loss should lead to quicker learning.

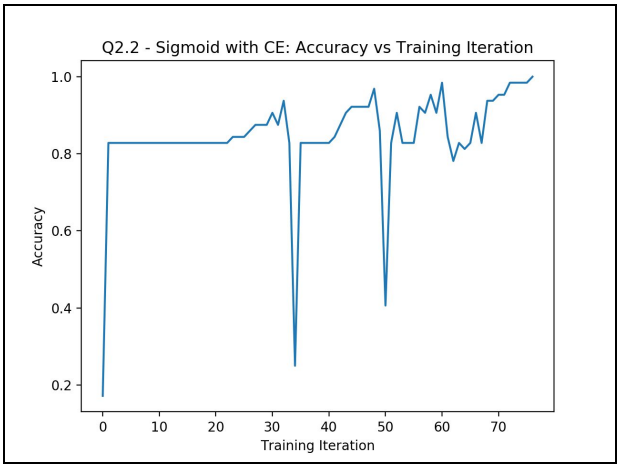
I predict that cross-entropy loss is a better loss function for classification (binary classification in this homework), whereas L2 loss is better for regression.

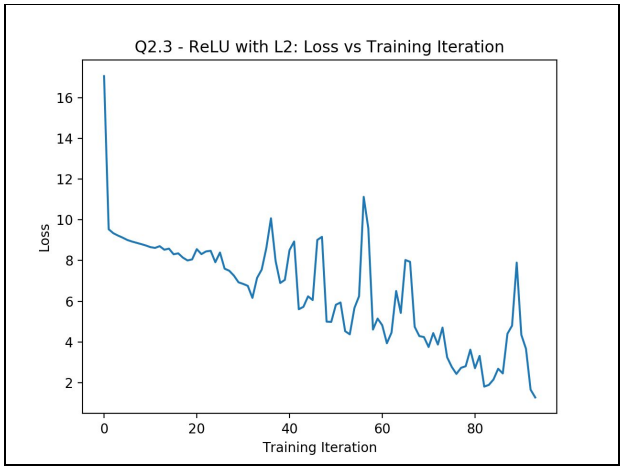
2 Fully Connected Network

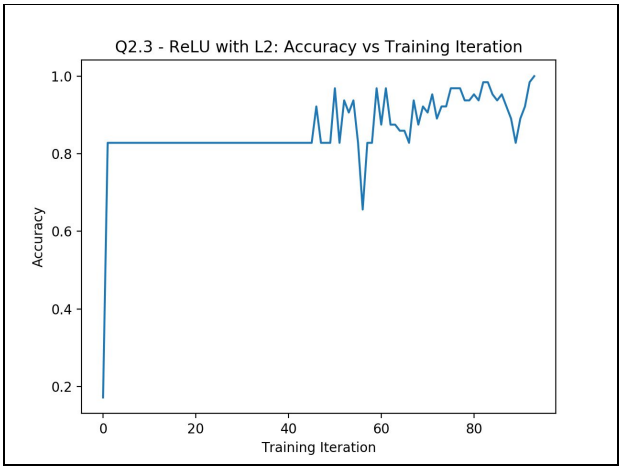


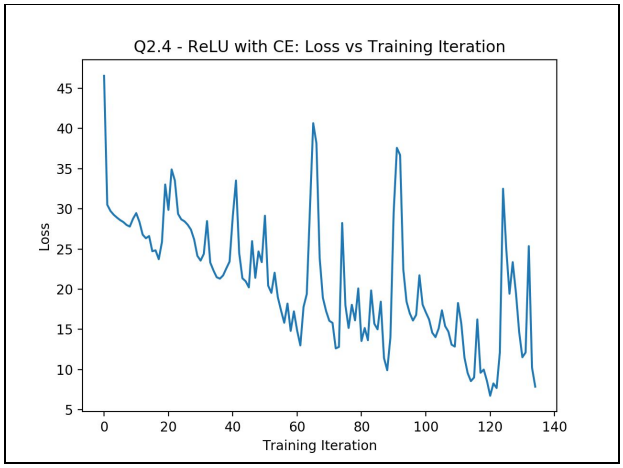


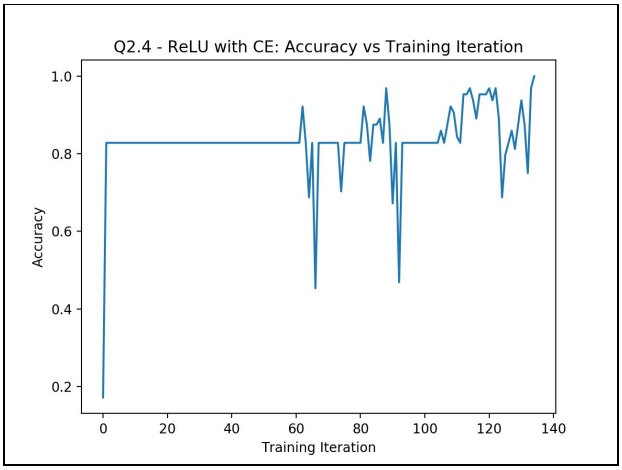












Model	**Iterations to reach 100% accuracy**
Sigmoid with L2, lr=0.1	147
ReLU with Cross-Entropy, lr=0.05	138
ReLU with L2, lr=0.1	94
Sigmoid with Cross-Entropy, lr=0.1	77

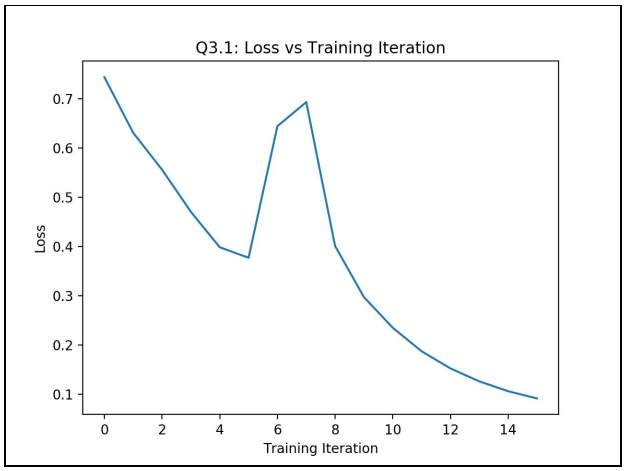
*Note that all models in this question use a learning rate of 0.1 except for ReLU with cross-entropy loss which uses rate 0.05. Using a learning rate of 0.1 for ReLU with cross-entropy resulted in the model not converging, even after 10,000 iterations.

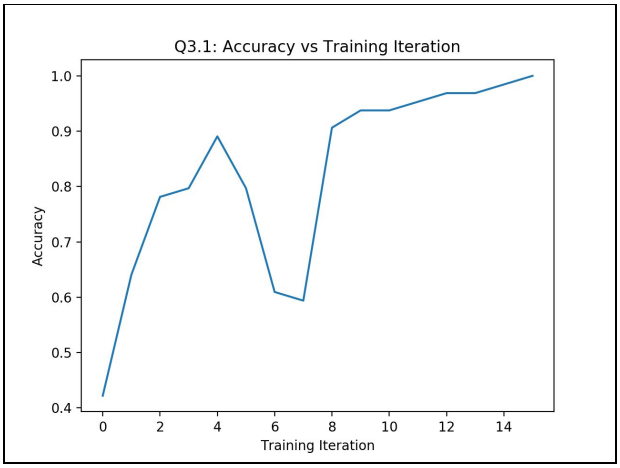
According to our experiments, the two fastest models to reach convergence were the ReLU activation function with L2 loss and the sigmoid activation with cross-entropy loss.

The overall fastest model was the sigmoid activation with cross-entropy loss. This makes sense, since we predicted in question 1 that cross-entropy loss would be superior to L2 loss for classification. Similarly, it makes sense that sigmoid outperforms ReLU for binary classification, because sigmoid maps to a value between 0 and 1, whereas ReLU simply maps to a positive number.

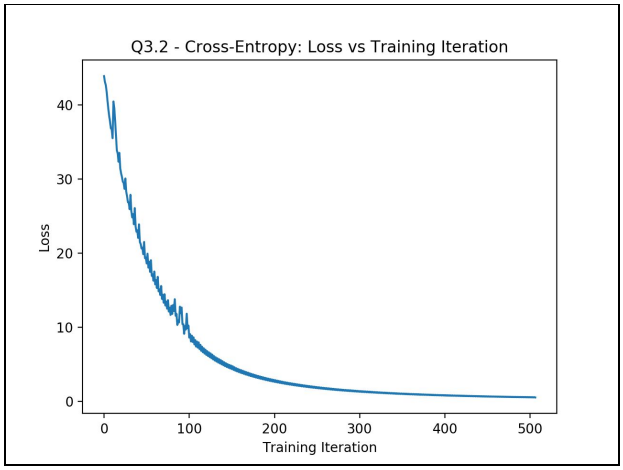
Our experiments indicate that for the sigmoid activation, cross-entropy loss is superior to L2 loss. This makes sense, as the sigmoid activation is essentially mapping to a probability/confidence that a given example is a 0 or a 1. Using the cross-entropy loss function, probabilities that are far off will be *heavily* penalized since the negative log of very small number is relatively high. For L2 loss, however, very wrong predictions aren't penalized that heavily, as the difference in mean-squared distance between the true label and predicted label for two incorrect predictions won't vary by as much.

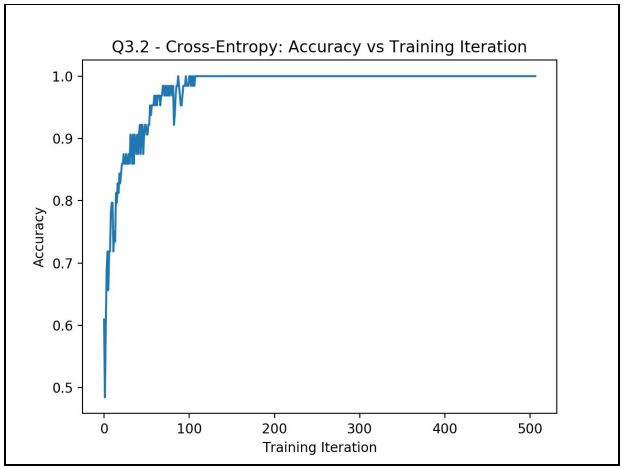
3 Convolutional Network

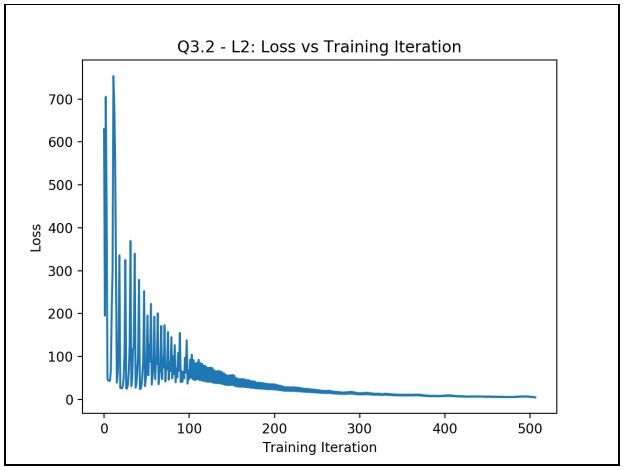


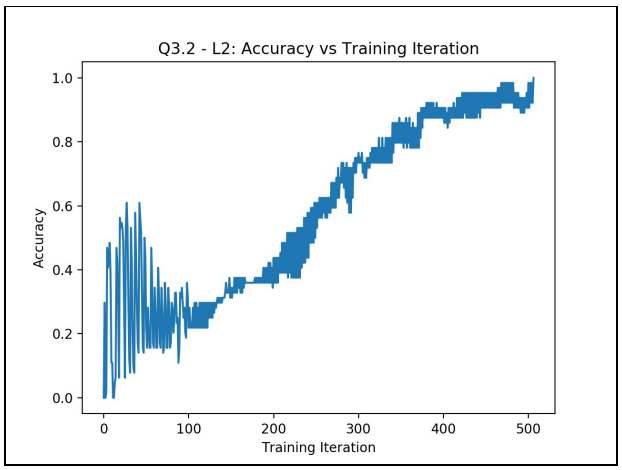


Using the line dataset, we construct a network with two convolutional layers and one fully connected layer, concatenated with a cross-entropy loss and learning rate 0.1. Compared to our results from question 2, we see that convergence is achieved *much* faster, in only 16 iterations. This makes sense as the line dataset encodes spatial relations. That is, examples with label 0 correspond to squares on the left side of the line, whereas examples with label 1 correspond to squares on the right side of the line. The kernels used in convolutional networks are specialized for capturing spatial relations, thus it makes sense that our convolutional network would converge extremely fast.









Using the detection dataset, we build on the network from question 3.1, adding another fully connected layer on top of the convolutional layer. We use learning rate 0.01 for the binary classification fully connected layer, and learning rate 0.001 for the width detection fully connected layer.

We note that this network takes the longest to converge, at 507 iterations. The first fully connected layer (for binary classification), does achieve 100% accuracy after only 130 iterations, however. This makes sense, as the width detection task is more difficult as the network must predict values between 1 and 8, as opposed to just binary classification. Further, it makes sense that this network overall takes the longest to converge. The convolutional layers of this network need to achieve two tasks: both classifying shapes and predicting widths. Since the convolutional layers need to complete both tasks, they must be general enough to provide useful information to both fully connected layers.