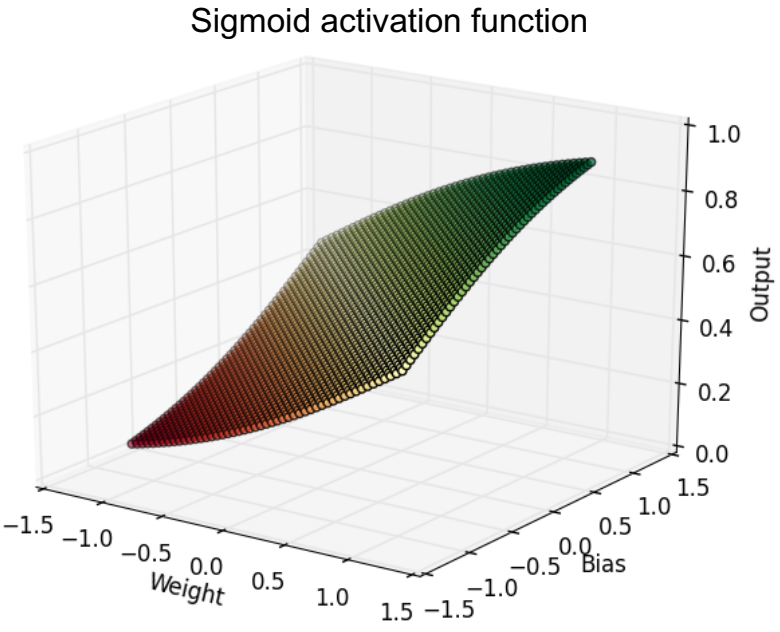
Project 4B Report
Group 24: Ariana Familiar, Eric Quesada, Prateek Singhal
Dec 2017

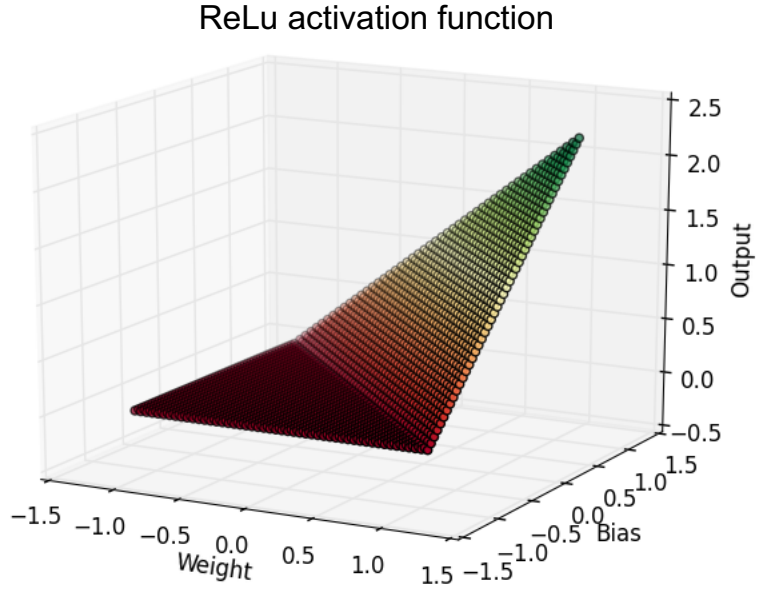
Part 1

1.

a.

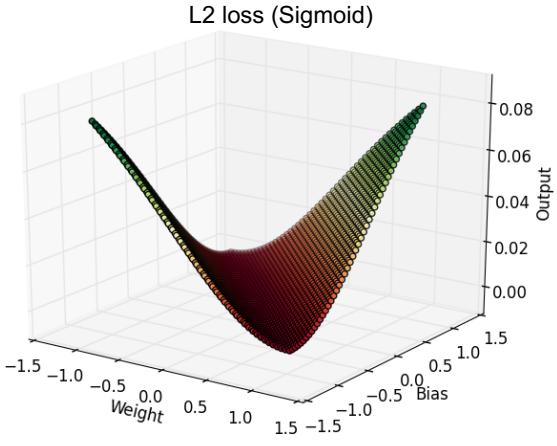


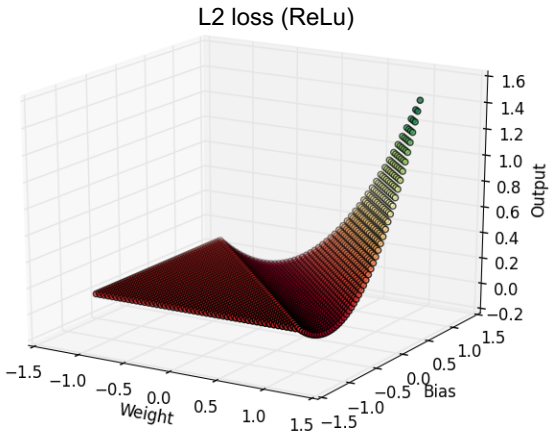
b.



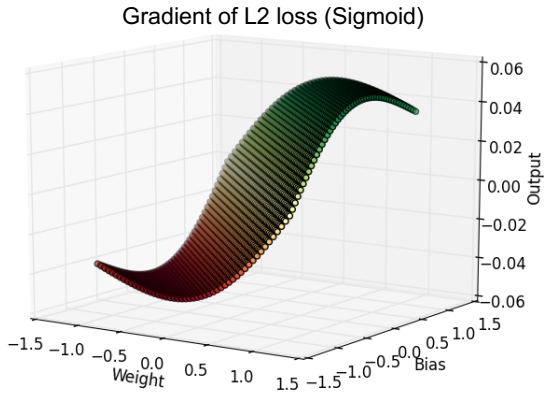
2.

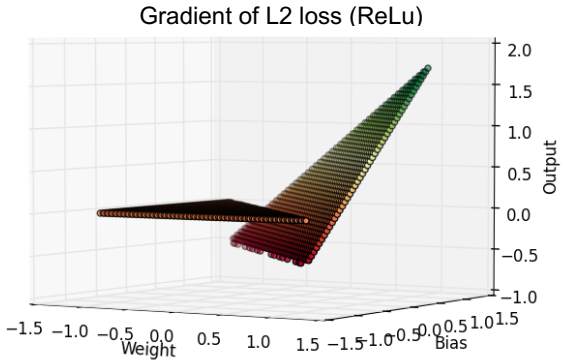
a.


L2 loss (Sigmoid)
Weight
Bias
Output


L2 loss (ReLu)
Weight
Bias
Output

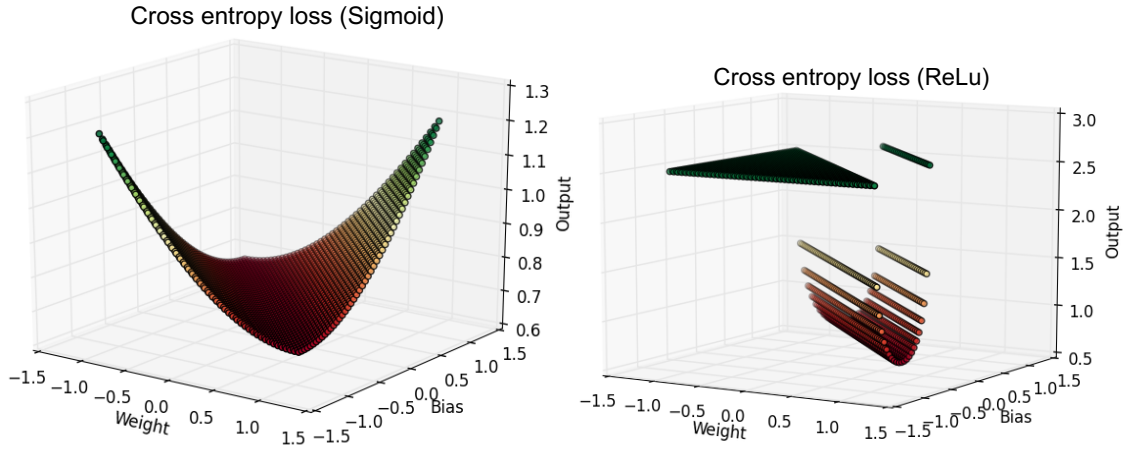
b.


Gradient of L2 loss (Sigmoid)
Weight
Bias
Output

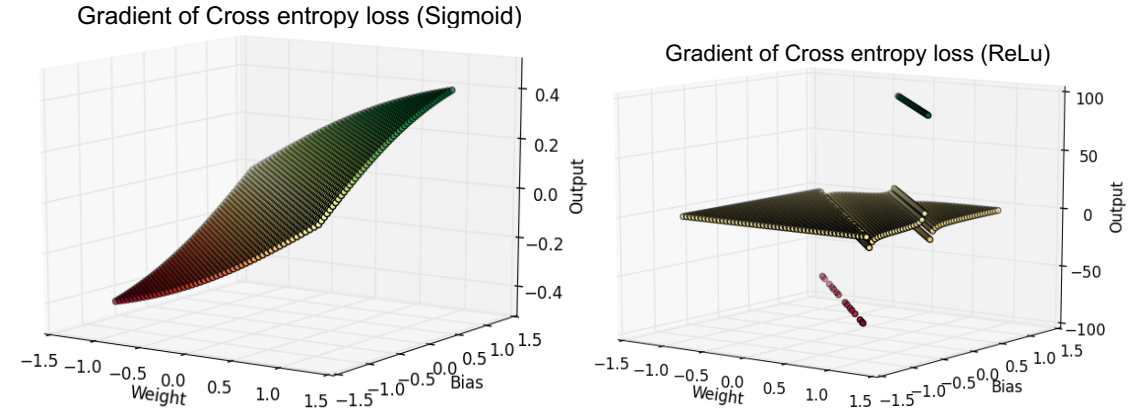

Gradient of L2 loss (ReLu)
Weight
Bias
Output

3.

a.



b.



4.

a. The main differences between L2 and cross entropy loss is that the cross entropy function aims to find the difference between probability distributions, and so is not best to calculate cost for output values that cannot be described as probabilities, while L2 calculates the mean squared error between the expected and actual output values. L2 seems appropriate when the output values are expected to be continuous and normally distributed (as in the relu layer output), while the cross entropy loss is more suitable when the output is nonlinear (as in the Sigmoid function).

b. When using a sigmoid activation function, the main difference between the gradients from L2 loss and cross entropy loss is that the L2 gradient calculation involves the derivative of the sigmoid function, while the cross entropy gradient calculation does not necessarily involve the derivative. Using the combination of L2 loss and sigmoid functions proves to be an issue, as the L2 gradient at the

upper and lower range of values trends towards 0 (i.e. begins to “vanish”). Specifically, when the error of the output is greatest, the slope of the sigmoid function, and thus the derivative, becomes small, causing the L2 gradient and resulting backpropagated error to be small as well (i.e. weights with large errors will not be changed much). Instead, the cross entropy gradient is a better match to the sigmoid function, and is a good model of the error of the sigmoid output (greater gradient when error is greatest).

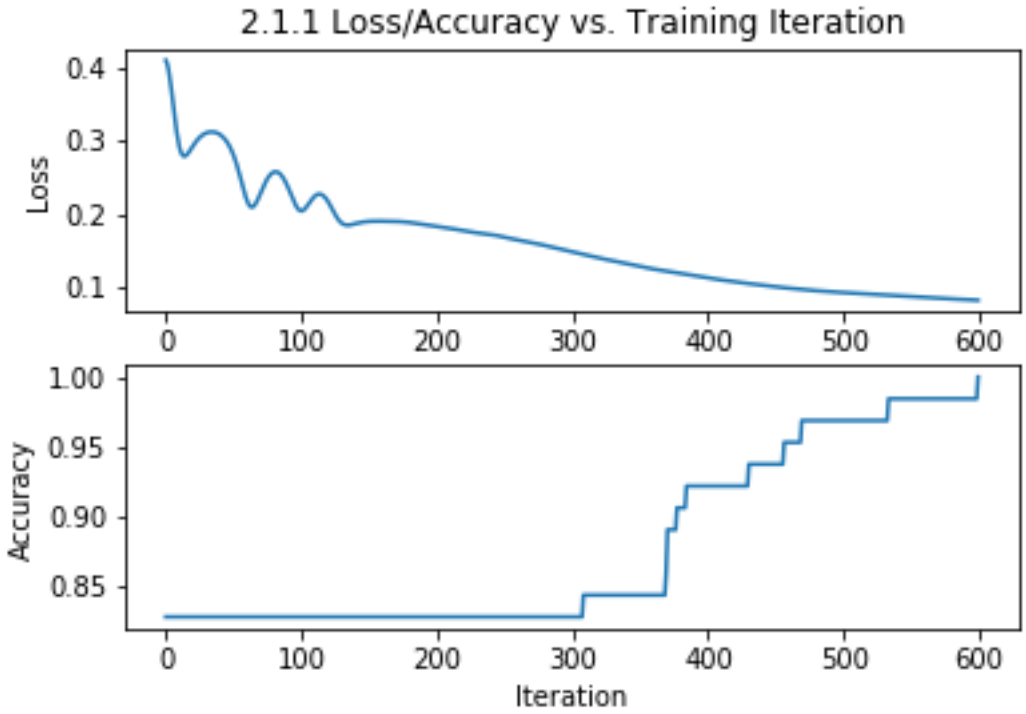
When using a relu activation function, the cross entropy loss function is more sensitive to the binary results of the relu derivative than the L2 loss function. Relu does not generate values that are well described as probabilities, thus the cross entropy gradient does not produce stable values that capture the magnitude of error produced by a relu function. The L2 gradient is more suitable for handling the continuous and linear nature of the relu output.

c. Learning would be problematic in the case of the sigmoid activation function with an L2 loss function, because the derivatives at extreme values are small, causing the gradient to be small, and so the weights with the greatest prediction errors would not be changed much in backpropagation. In this way, the combination of sigmoid activation and L2 loss functions generate a model that will learn very slowly (especially with poorly initialized weights). On the other hand, the cross entropy gradient does not involve the derivatives of the activation function, so learning would be much faster because a greater error will be assigned to weights with greater prediction errors.

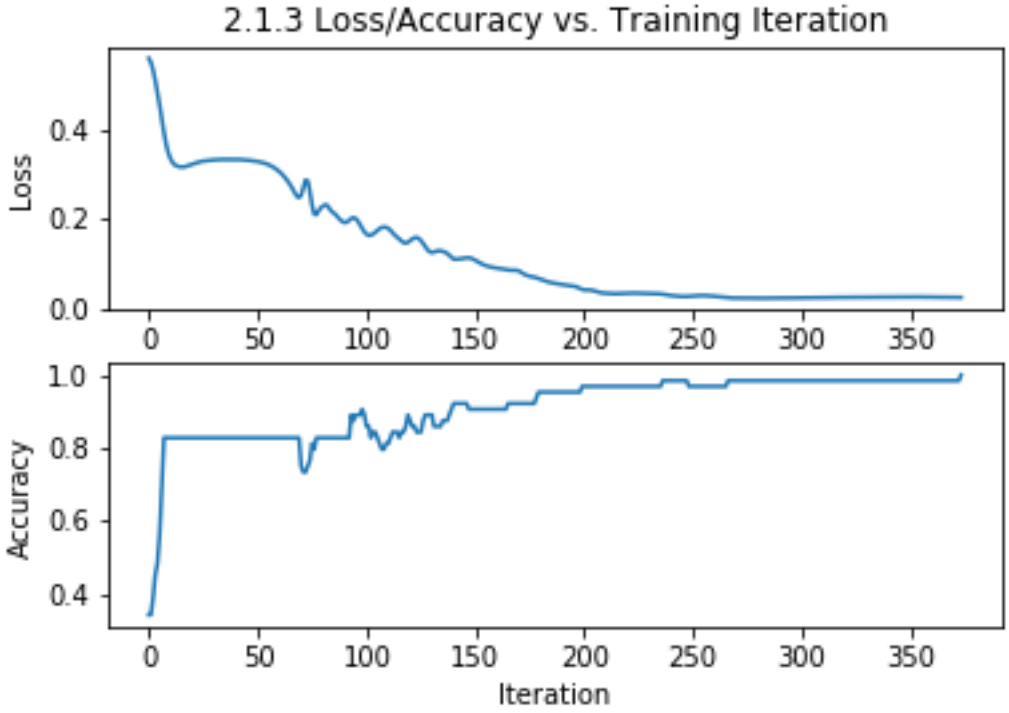
   For the relu activation function, an L2 loss function will lead to faster learning compared to a cross entropy cost function, as the cross entropy gradient is not stable in assigning error to the predicted values of the relu output. Instead, the L2 loss is a good match for the continuous and linear aspects of the relu output.

Part 2.1

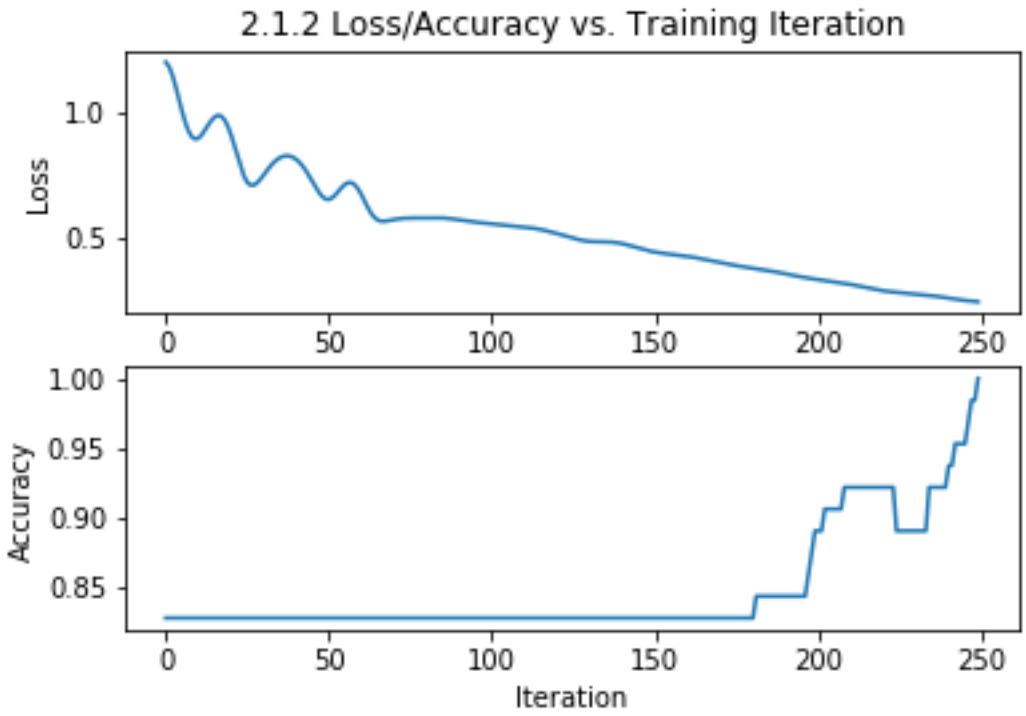
1. Sigmoid activation, L2 loss



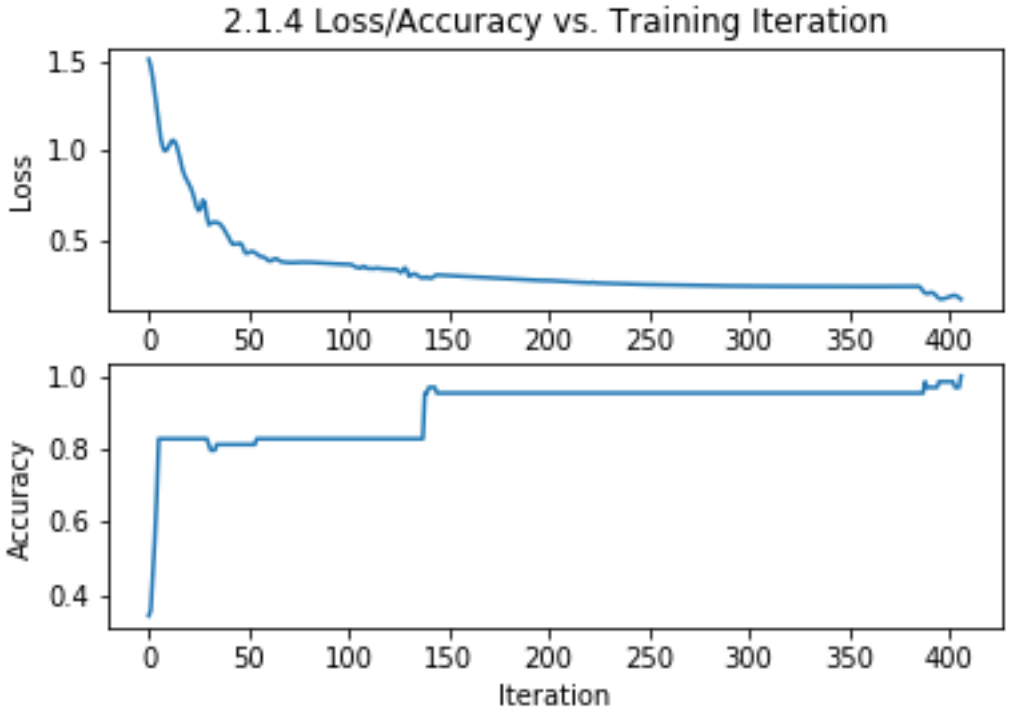
2. Sigmoid activation, cross entropy loss



3. Relu activation, L2 loss



4. Relu activation, cross entropy loss



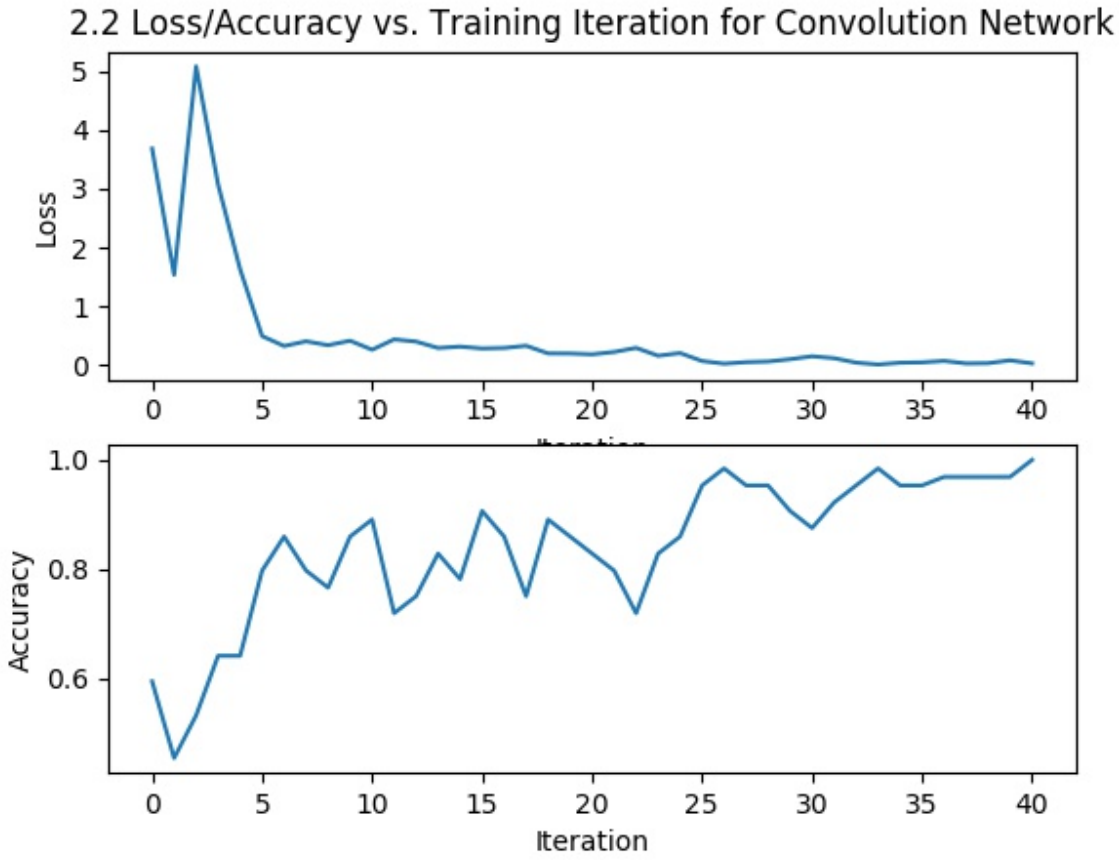
5. The order (in increasing order) is: 3 (relu/L2), 2 (sigmoid/cross entropy), 4 (relu/cross entropy), 1 (sigmoid/L2).

   The sigmoid and L2 loss combination leads to the slowest learning because the L2 gradient depends on the derivatives of the sigmoid function, which are lower at the tail ends of the sigmoid function. This results in a lower backpropagated error for weights of greater error (farthest from ground truth), and so these weights are not updated properly.

   The L2 loss performs better when the output values are expected to be continuous and linear, as in the relu output. This combination generated a model with the fastest learning.

   Cross entropy loss led to better learning with the sigmoid activation function compared to the relu, because cross entropy computes loss as a difference in probability distributions, and the relu output is not well described as probability values.

Part 2.2


2.2 Loss/Accuracy vs. Training Iteration for Convolution Network
Loss
Accuracy
Iteration