Assignment Report

The structure of the report is organized as follows:

For each implementation of the neural network, a detailed analysis is conducted on both the loss functions and the activation functions utilized within the model. Additionally, varying learning rates are applied to assess their impact on the model's performance, allowing for a comprehensive evaluation of which configurations yield the most optimal results. Through this approach, the report aims to provide a thorough comparison of the performance metrics across different combinations of activation functions, loss functions, and learning rates.

1.1. Two Layer Network

1.1.1. Loss Function and Activation Function

The performance of SSE and CE loss functions was analyzed. The observations were as follows.

1. A learning rate of 0.15 seemed to fare better in both cases as the costs converged faster with respect to the other learning rates. It was also interesting to observe that it still was the most effective learning rate for both the ReLU and Tanh activation functions as seen in figure 1.
2. All three activation functions performed the same based on the different between their loss functions based on figure 2 and 3. Also, SSE is the best performing loss function for this use case.

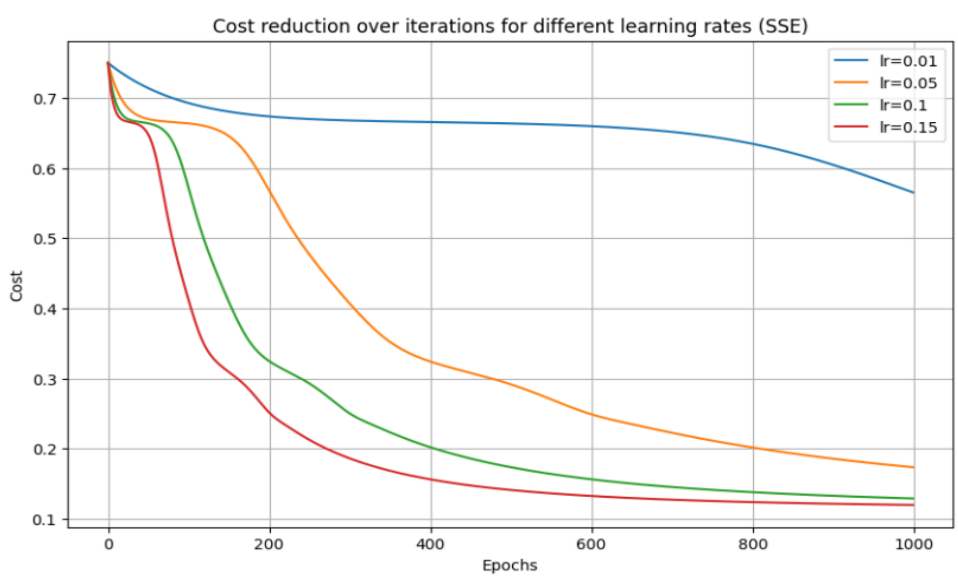

Figure 1: Cost reduction over iterations for different learning rates.

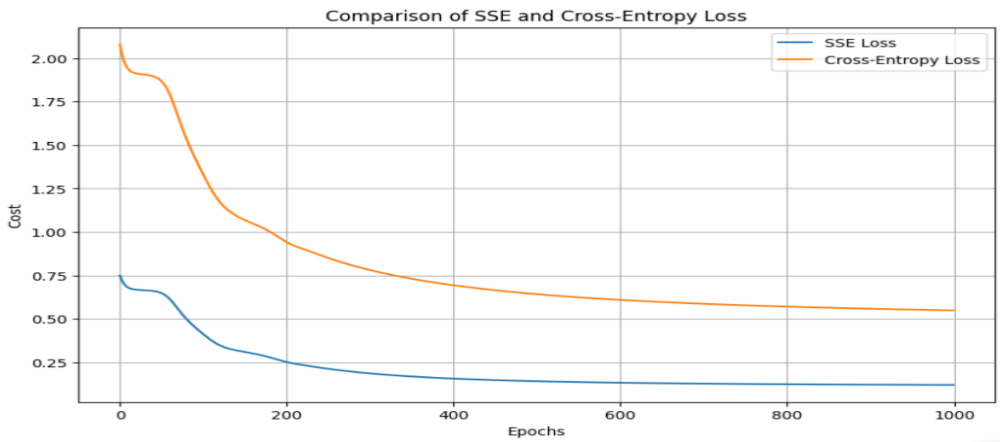

Figure 2: Comparison of SSE and CE

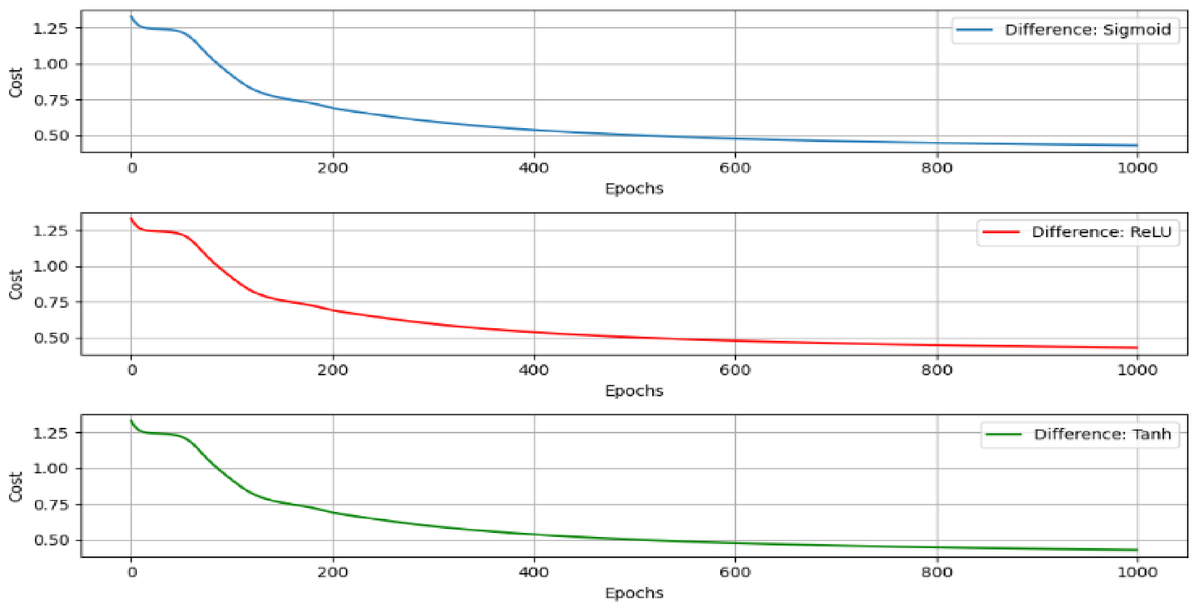

Figure 3: Loss Function difference across the three activation functions.

1.2. L-Layer Network

For this layer network, the number of layers is flexible and no longer restricted to just two layers. Here are some of the findings.

As always, the relationship between the loss and activation functions was being analyzed.

1. A learning rate of 0.15 seemed to still fare better in both cases as shown for a two-layer network. For example, the graph showing the cost function using both SSE and sigmoid, the learning rates do not experience any form of illustration as shown in a two-layer network (Fig. 4). Hence, the L-Layer structure is smoother than the two-layer network.

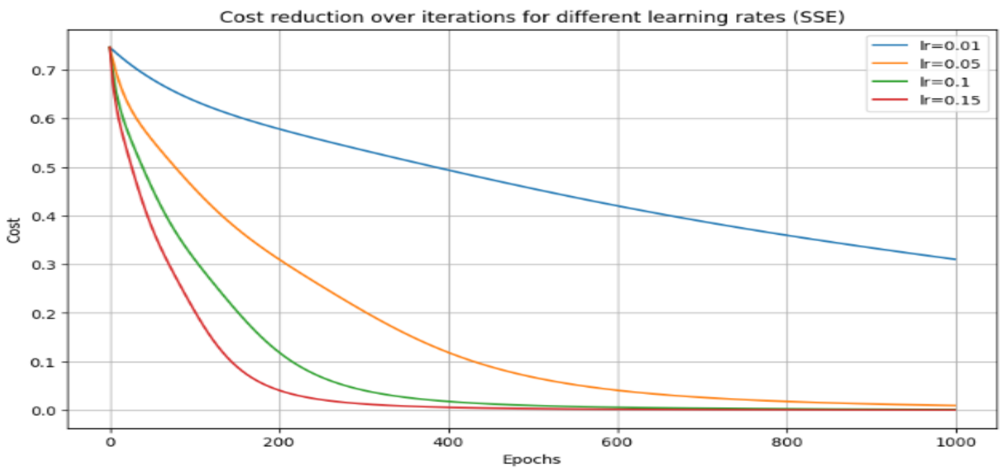

Figure 4: Cost reduction for L-Layer Network

2. The absolute difference between the SSE and CE for the L-Layer Network also appeared to be smoother that the Two-Layer Network. It converged more quickly as depicted in fig.5. It can be inferred for ReLU that it converges quickly compared to the rest.

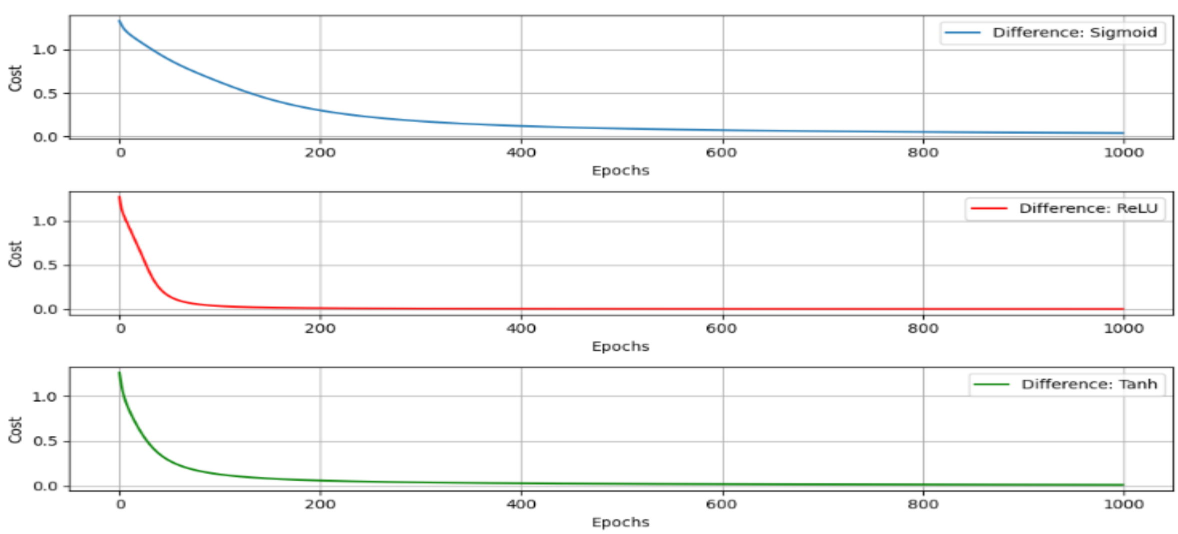

Figure 5: L-Layer Cost Difference

1.3 Keras Implementation

In the early implementations of the model, a sigmoid function was applied as the activation function in the hidden layers. However, for the experiments conducted in this section, both the tanh and ReLU activation functions were employed. The objective was to compare and evaluate the performance of these two activation functions under similar conditions. The analysis of their performance, behavior, and overall impact on the model's output is detailed as follows:

1.3.1 Using SSE as loss function:

1. The curve appeared smoother than an L-Layer Network.
2. Unlike the previous layer structures, the most effective learning rates were 0.1, 0.15 and 0.05 as shown in Fig.6 when using the sigmoid activation function.

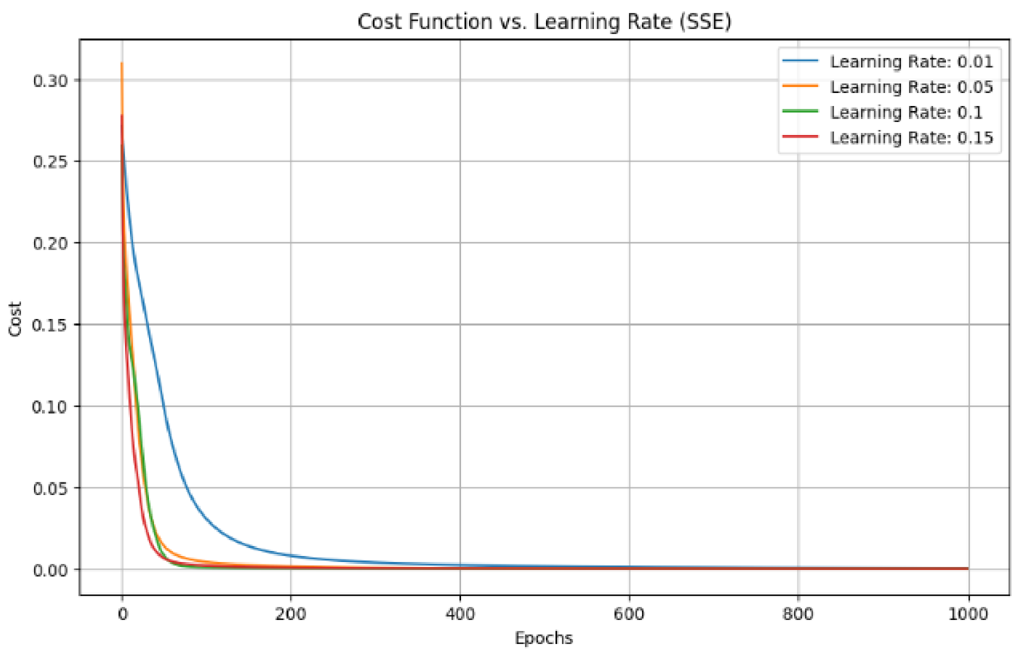

Figure 6: Using Sigmoid

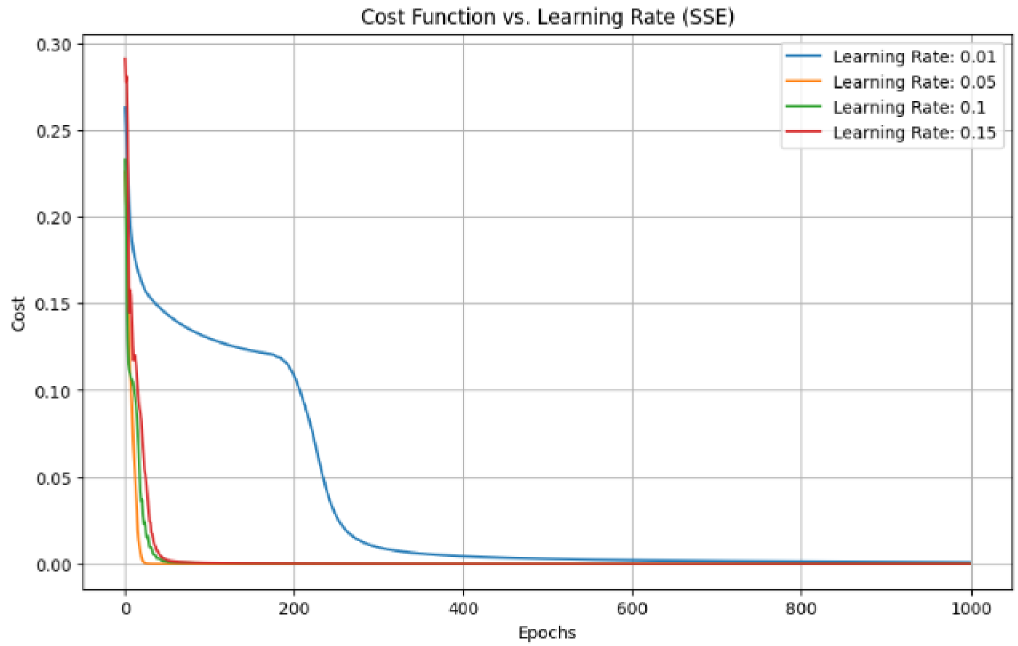

Figure 7: Using ReLU

3. For the ReLU activation function in fig.7, a different trend is noticed. A learning rate of 0.05 appears to be the most effective.
4. In fig 8, a learning rate of 0.15 appeared to be the most effective and the curve is also smoother than that of the L-layer and two-layer network.

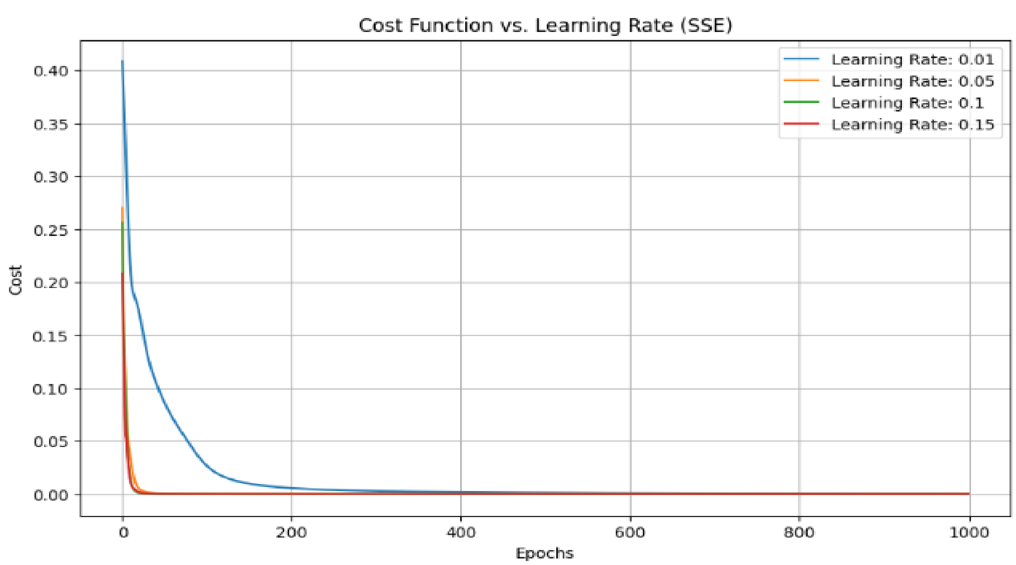

Figure 8: Using Tanh

1.3.2 **Using CE as loss function:**

1. The curve appeared smoother than an L-Layer Network.
2. Like that of the SSE loss function when using a sigmoid function as depicted in fig. 9.

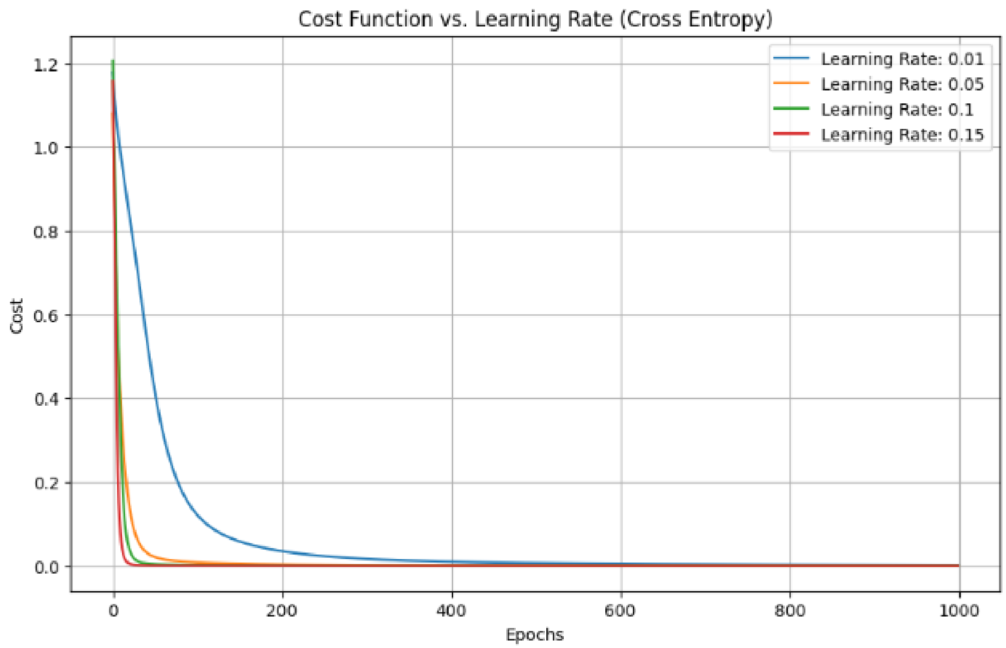

Figure 9: Using Sigmoid (CE)

3. Learning rates of 0.15 and 0.1 prove the most effective rates for the cost when using tanh activation function as shown in fig. 10.

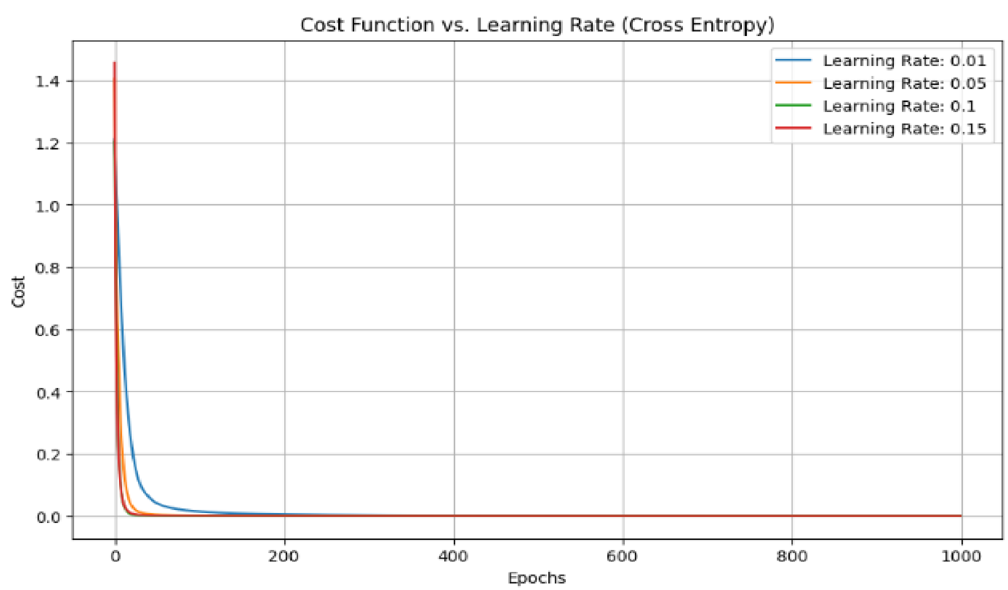

Figure 10: Using Tanh (CE)

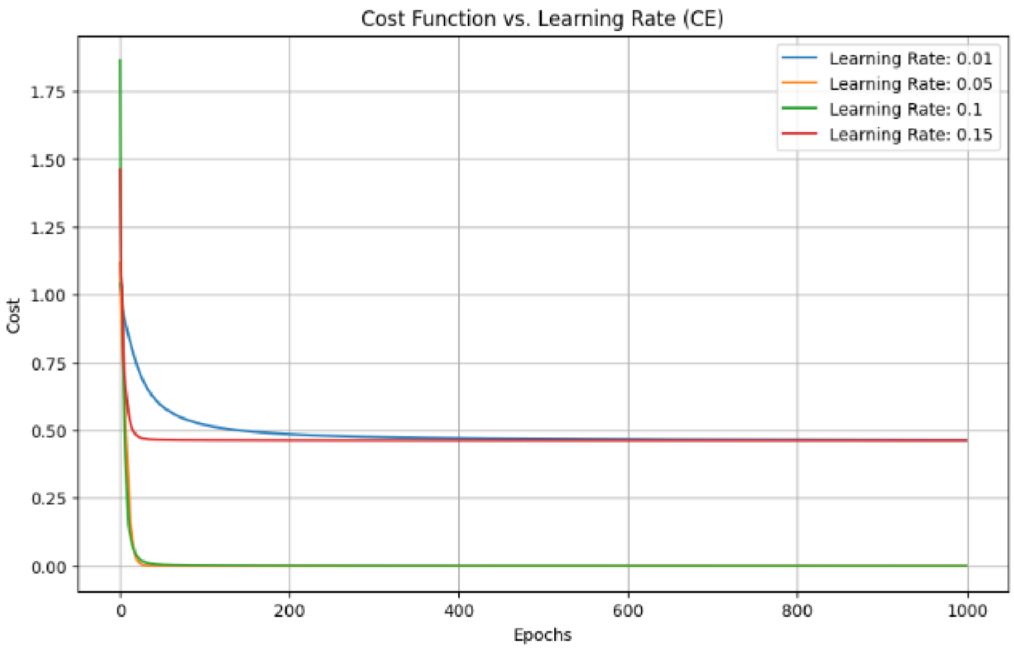

Figure 11: Using ReLU (CE)

4. Using the ReLU function, we obtain also obtain a smooth curve as exhibited in the other functions. However, a learning rate of 0.15 flattens out pretty soon as shown in fig. 11.