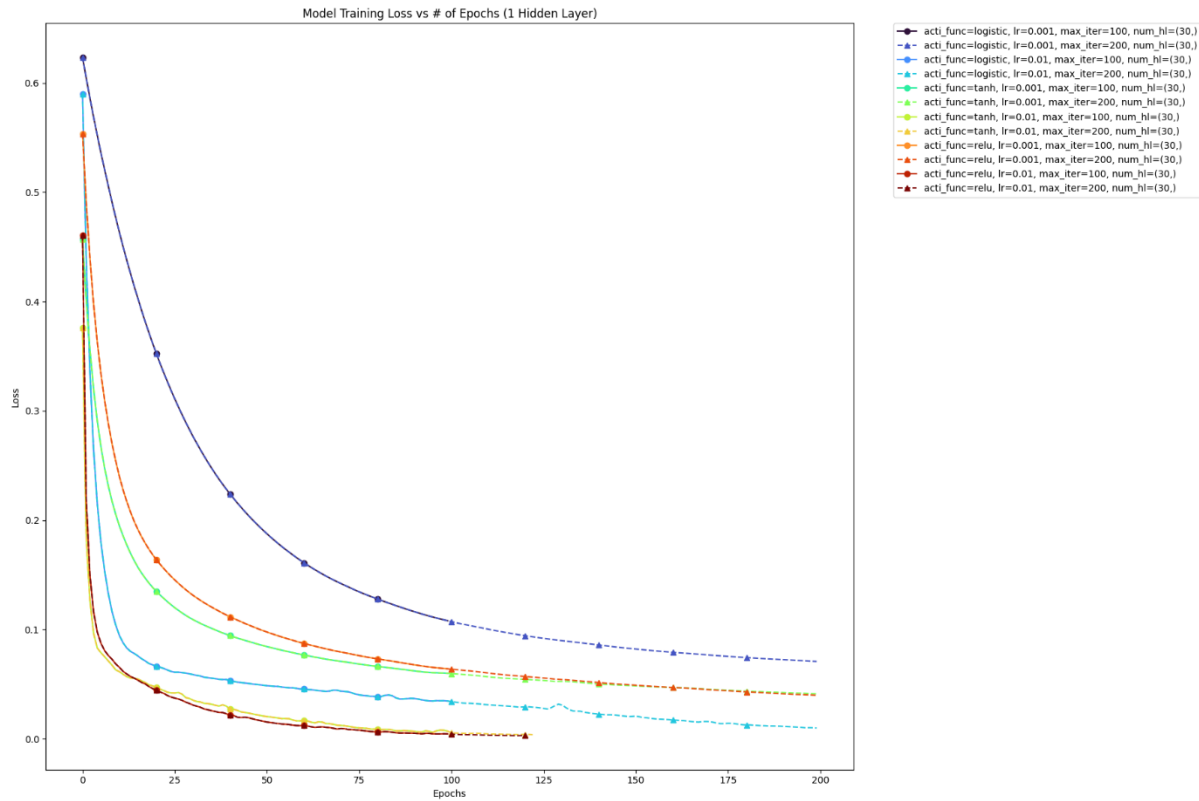
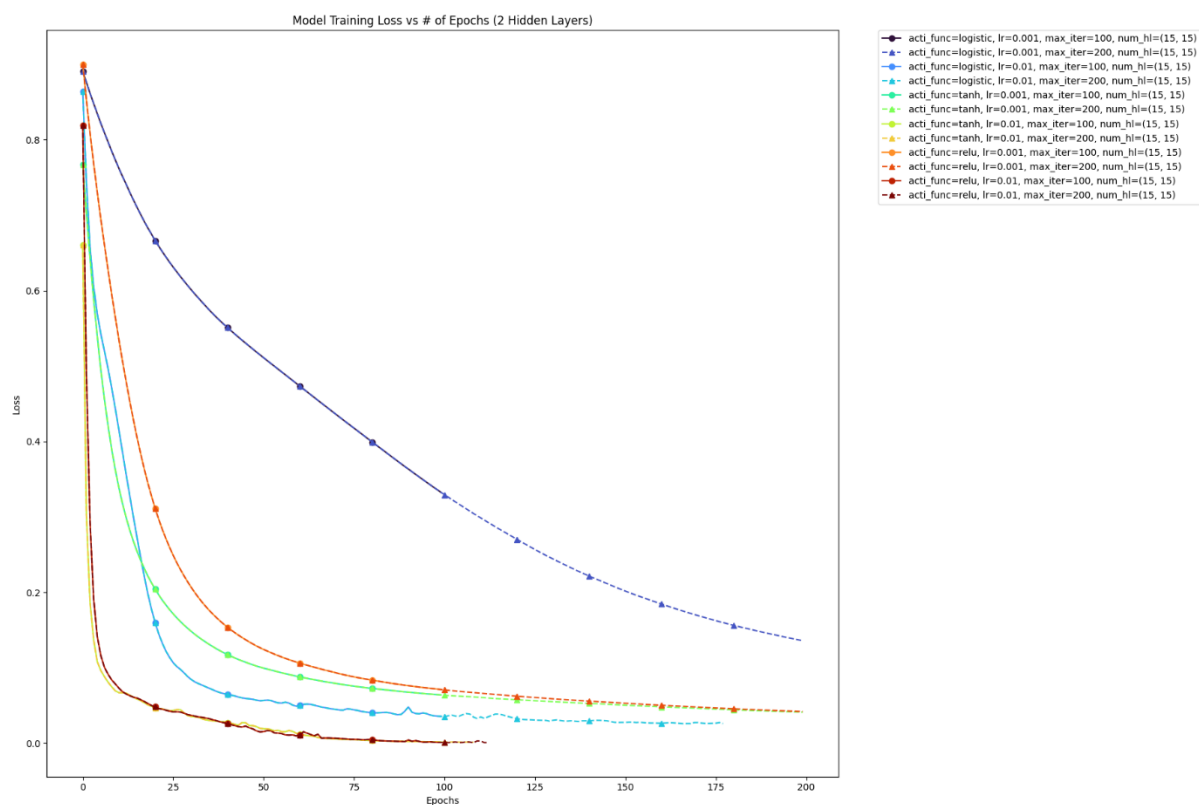


1 Generated Plots

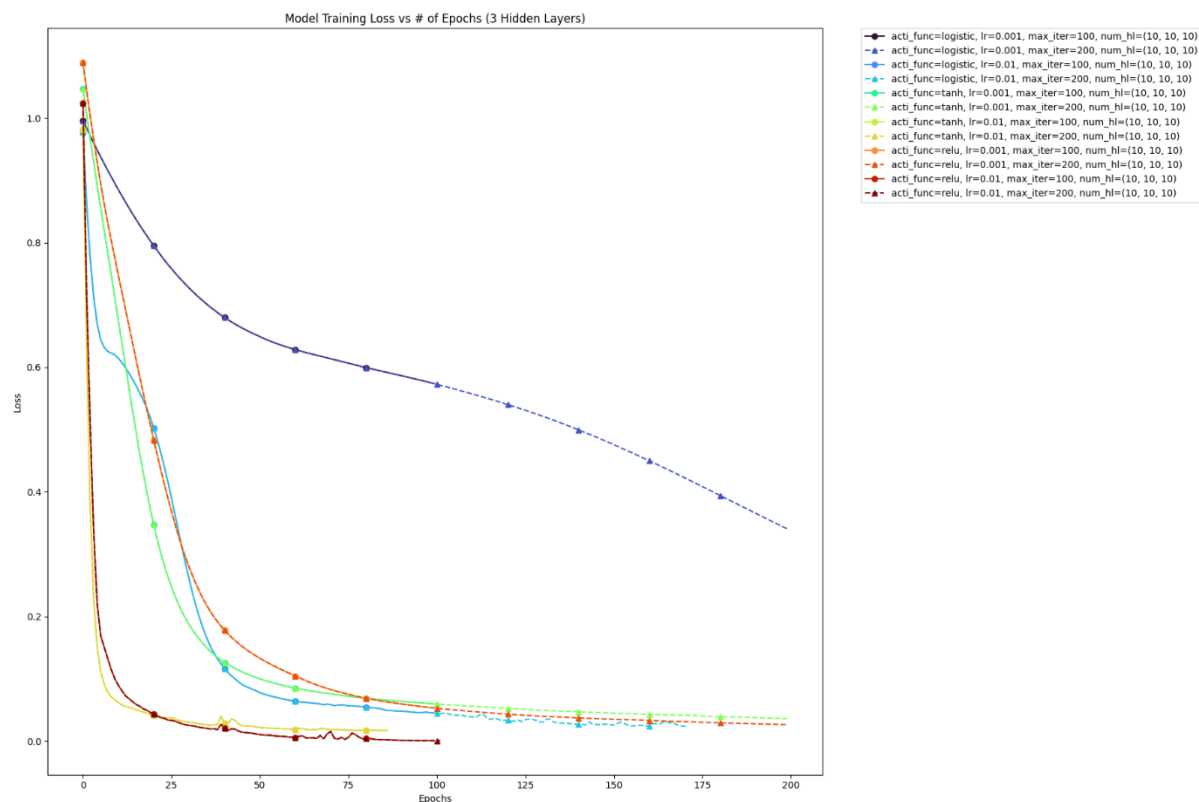
1.1 Plot 1, 1 Hidden Layer Results



1.2 Plot 2, 2 Hidden Layers Results



1.4 Plot 3, 3 Hidden Layers Results



2 Hyperparameter Output Log

2.1 Table 1, Tuning Results

activation	learning_rate	max_iter	layers	train_loss	test_loss
relu	0.001	100	(10, 10, 10)	0.053244325	0.052922405
relu	0.001	200	(30,)	0.039798904	0.057209095
tanh	0.001	100	(30,)	0.059895359	0.057294759
logistic	0.001	200	(30,)	0.070714478	0.057412753
tanh	0.001	200	(30,)	0.041130591	0.05842713
relu	0.001	200	(15, 15)	0.04194449	0.058835607
tanh	0.001	100	(15, 15)	0.063924738	0.063827305
relu	0.001	200	(10, 10, 10)	0.02640598	0.06458877
tanh	0.001	100	(10, 10, 10)	0.059840809	0.065945362
relu	0.001	100	(15, 15)	0.071157575	0.066145788
relu	0.001	100	(30,)	0.063992531	0.067218317

tanh	0.001	200	(15, 15)	0.041165367	0.073826107
tanh	0.001	200	(10, 10, 10)	0.036063493	0.081769474
logistic	0.001	100	(30,)	0.107972367	0.089249761
logistic	0.01	100	(30,)	0.034531143	0.095079409
logistic	0.01	100	(15, 15)	0.035598637	0.107402526
logistic	0.01	100	(10, 10, 10)	0.045381151	0.112124291
logistic	0.01	200	(30,)	0.009824413	0.113062238
logistic	0.01	200	(10, 10, 10)	0.024457083	0.119210024
logistic	0.001	200	(15, 15)	0.135929229	0.122607644
logistic	0.01	200	(15, 15)	0.026033669	0.126303559
tanh	0.01	100	(30,)	0.006873298	0.12825896
tanh	0.01	200	(30,)	0.003734419	0.128540738
tanh	0.01	100	(15, 15)	0.001598358	0.129740702
tanh	0.01	200	(15, 15)	0.001298555	0.13351317
tanh	0.01	200	(10, 10, 10)	0.017116777	0.145839509
tanh	0.01	100	(10, 10, 10)	0.017116777	0.145839509
relu	0.01	100	(30,)	0.004441483	0.145974986
relu	0.01	200	(30,)	0.002960902	0.150197622
relu	0.01	100	(10, 10, 10)	0.000716148	0.160982376
relu	0.01	200	(10, 10, 10)	0.000664295	0.162373613
relu	0.01	100	(15, 15)	0.001071924	0.201495023
relu	0.01	200	(15, 15)	0.000847754	0.257427894
logistic	0.001	100	(15, 15)	0.332627411	0.314994334
logistic	0.001	200	(10, 10, 10)	0.340525656	0.332368282
logistic	0.001	100	(10, 10, 10)	0.57402449	0.570972265

3 Analysis

3.1 Summarization of Results and Changes to Source Code

There are a few alterations that we made to the source code. Firstly, we removed the primary comments and added ones explaining each step. We kept the main framework and hyperparameters (with slight alterations) as

well as expanded on the layer size testing. Additionally, we made it so that convergence warnings were ignored and not printed during execution. We can see that the best learning rate was 0.001 and out of those, the best activation functions were the 'relu' and 'tanh' activation functions. Most notably, the 'relu' activation function with (10, 10, 10) hidden layers and a learning rate of 0.001 gave us the best test loss with 0.052922405. The next best test losses gave similar results with just a single hidden layer of 30 neurons (with varying activation functions). The next best ones used 2 hidden layers with 15 neurons each. Overall, the 'relu' activation function performed best with 3 hidden layers. We think this is because the 'relu' activation function helps with vanishing gradients when using multiple hidden layers.