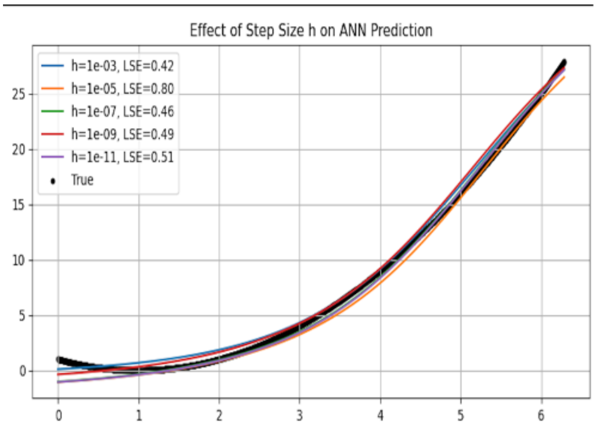
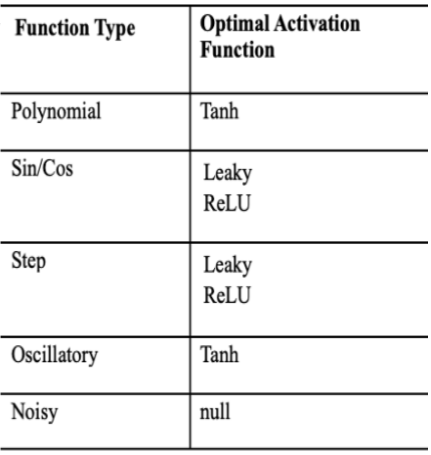
The Feed-forward ANN developed in this project utilized an approach driven by methods seen in class, with some slight modifications. After creating the structure of the neural net based on neurons (n) and hidden layers (N), the methodology used the central difference method to numerically compute the gradient of the least square error function for each x and y of the sampled data point$\nabla E(x) \approx (E(x + h) - E(x - h)) / (2h)$. The objective was to minimize the error function by solving the nonlinear system of equations $\nabla E(x) = 0$ where the vector x represents the weights and biases of the ANN. Within this method, Gauss-Seidel iteration solved the system $A\,y = -f(x)$ at each iteration with the goal convergence based on $||y|| < tolerance$. Broyden's iterative method was chosen over Newton's method to avoid recomputing a Jacobian matrix at each iteration, which was observed to be numerically unstable and computationally expensive.

Optimization of the method parameters was performed on the optimal step size, h, for the central difference and Gauss-Seidel method through systematic testing, which yielded a step size of 1e-3 as optimal. The results (figure 4) showed that an excessively large or small step size led to inaccurate approximations and reduced stability. A similar test was performed on the relationship between transfer function and sample function type (table 2). Both tests were evaluated based on minimum LSE.





Function Type	Optimal Activation Function
Polynomial	Tanh
Sin/Cos	Leaky ReLU
Step	Leaky ReLU
Oscillatory	Tanh
Noisy	null

Figure 4 *Table 2*

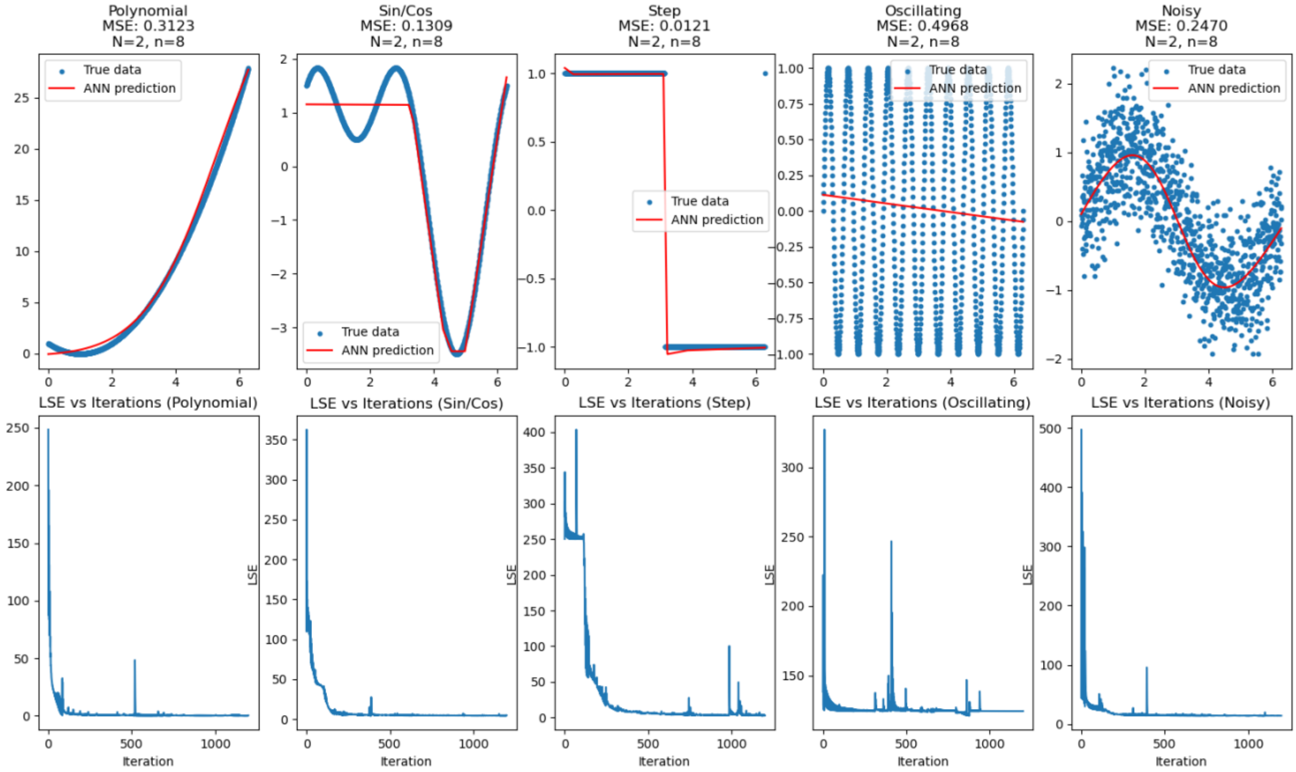
A major challenge during training was controlling the numerical instability that arises with this method, specifically an exploding least squares error. This was addressed by clipping the central difference gradient if it was too large in magnitude, normalizing the input data between [0,1] and introducing a normalization constant to prevent division by very small numbers, so that $update = [\Delta f - A\,y] / (||y||^2 + 1)$. Due to the importance of the initial guess on quasi-newton methods, a retry mechanism was implemented in broydens that reset the vector x to a new initial guess and the Jacobian approximation to the identity if the LSE had exploded.

The performance of the method was evaluated on ease of implementation, approximation accuracy, convergence behaviour and computational efficiency. This method was moderately difficult to implement, as it required the proper handling of numerical instabilities throughout iterations. It is very computationally intensive due to the repeated gradient evaluations; system of equation solves and iterative matrix update. The complexity of this method was determined to be O(I(N * n^2)), where N is hidden layers, n is neurons, I is iterations; it was observed that an increase in N and n greatly increased the computational time.

The approximation accuracy was evaluated using MSE to avoid bias from small inaccuracies between the sampled points and the points approximated by the ANN, as seen in figure x. The ANN showed strong approximation accuracy for smooth functions like the polynomial, while it struggled with high-frequency oscillations and sharp corners. The step function was smoothed near the discontinuity and the noisy function was handled reasonably well by capturing the general shape. These results become more accurate with more iterations and a better suited network structure for each type of function, however this greatly increases the amount of computational power required.

Convergence behaviour (figure 5) was seen to be initially very steep with oscillatory LSE values that trended downwards with occasional spikes. This can be explained by the nature of the broydens scheme, which may have “overshot” the error minimum causing fluctuations. The previously mentioned strategies greatly helped to reduce these issues. More traditional methods, such as gradient descent, or implementing line search and damping in our Broyden scheme may help to correct these issues [1].

Figure 5 –Results Visualized for 1000 Sample Points, Max Iterations of 1200



Future work should include more model optimization, namely for network parameters and the initial guess. The initial guess is set to be random * 0.1 and it would be useful to explore options tailored to a transfer function, such as Xavier [1]. Furthermore, the introduction of features such as line search or damping to improve the efficiency, convergence and approximation accuracy of the broyden's scheme.