CSI 5340 – Deep Learning and Reinforcement Learning

Assignment 1

Student Name: Nathaniel Bowness

Student Number: 7869283

Due Date: September 29th, 2023

This report will analyze and describe some interesting patterns seen while performing the outlined programming assignment to explore the fitting and generalization of regression models. Each section will outline the data used to generate the graphs, including any hyperparameters set to a constant value.

Polynomial Complexity and Noise:

Increasing the polynomial complexity increases the complexity of the function used to fit the given dataset. All graphs created use the following regression model settings, with fluctuating details represented on the graph axis, including Mean Square Error and Polynomial degree.

- $d \in \{1,\ 2, 4, 8, 16, 32, 64\}$
- $\sigma^2 \in \{0.05, 0.2\}$, one value used per line plotted.
- $N\ \in \{5, 50, 200\}$, one value is used and fixed in each graph.
- $M = 50$
- $Learning\ Rate = 0.01$
- $Iterations\ of\ GD\ =\ 2000$

Results Without Regularization:

Without regularization, the results show that as the polynomial degree increases, the $\overline{E_{in}}$ decreases regardless of the noise in the data. This confirms what was seen in class: the more complex the polynomial is, the better it can fit the dataset. Figure 1 a) shows that over-fitting occurs when the polynomial degree is large but the dataset is small. Once the degree of the polynomial with respect to the dataset is large, the generalization of that function is poor and leads to a high $\overline{E_{out}}$ values.

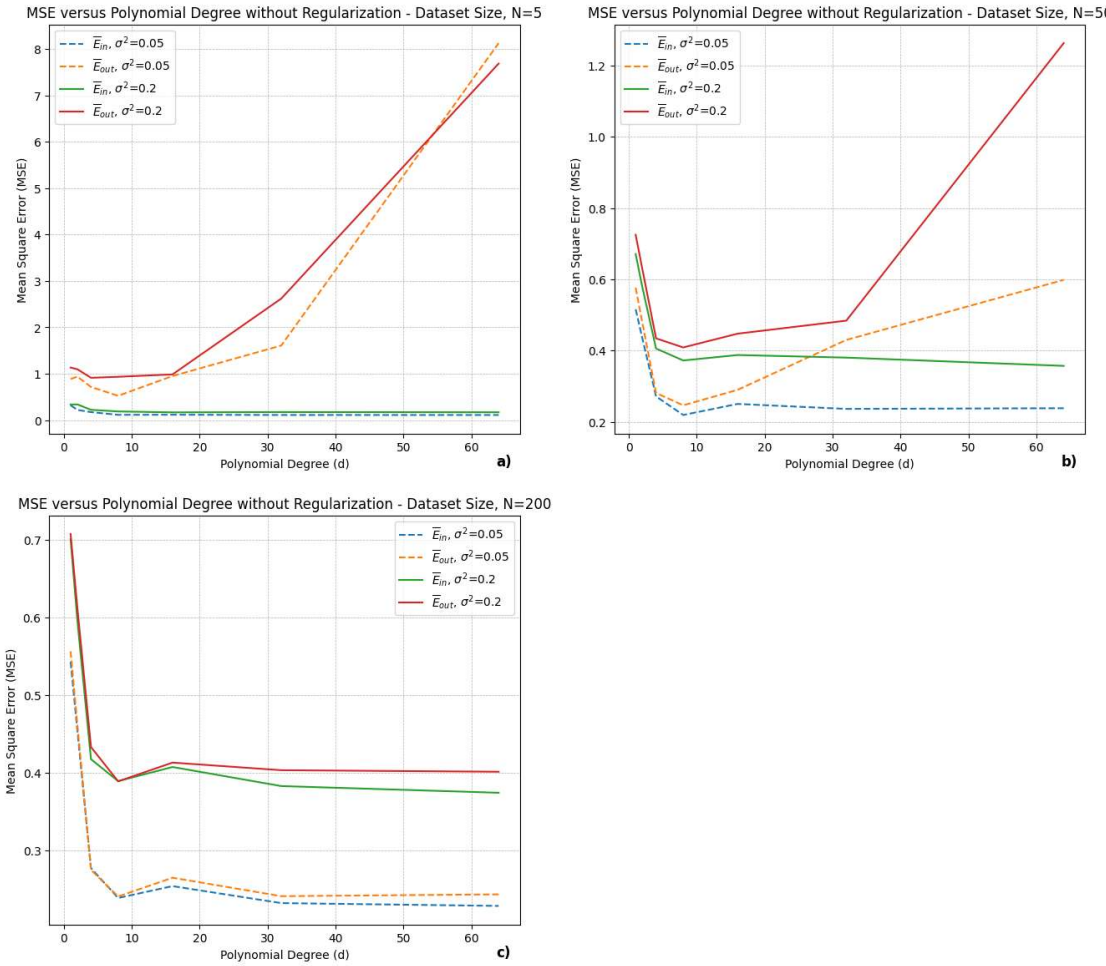

Figure 1: *Mean square error plotted against the polynomial degree of the regression model. Each figure is plotted against a fixed dataset size N and plots the data for* $\overline{E_{in}}$, $\overline{E_{out}}$, *at a specified variance values.*

From Figure 1b), we can see the under-fitting process turned to over-fitting on the dataset of N=50. The MSE decreases until around degree 8. At degree 8, the polynomial is ideal for that training dataset size when only considering the degrees plotted. After degree 8, the function over-fits the dataset, increasing the MSE. From Figure 1c), we can again see the same over-fitting to under-fitting pattern at around degree 8. From Figure 1 c), we can also see that with a larger dataset, a large polynomial of degree 64 still generalizes okay to the data with the value of $\overline{E_{in}}$ and $\overline{E_{out}}$ being similar. This shows that polynomials of larger degrees will need more data to properly generalize on other datasets. Another piece of data seen, which is not prominent, is a "double descent" curve in the plotted graph. At polynomial size 16, we can see another peak in the middle of the dataset. This may be a coincidence from the data, but it does show similar characteristics to the double descent pattern demonstrated in class.

Affect of Noise:

The effect of noise is visible in Figure 1b). The MSE of the $\overline{E_{in}}$ and $\overline{E_{out}}$ with a noise level of 0.2, is much higher than a noise level of 0.05. This trend of a higher MSE can also be seen in Figure 1c) and most of the graphs shown in this report. This higher value can be explained because the higher noise level leads to more irregular data that is harder to fit with a single function and predict accurately.

Results With Weight Decay Regularization and Comparison:

Weight decay regularization in gradient descent is used to help with the generalization performance of the machine learning model. For this assignment, the regularization should help prevent overfitting and reduce the impact of small changes in the data on the model. It was implemented with the following settings:

- $Weight\ Decay\ Value = 0.1$

The same type of plots shown in Figure 1 can be seen in Figure 2 but with regularization applied to the model. With regularization the $\overline{E_{in}}$ value also tends to decrease as the model's polynomial degree increases. Similarly, the noise level of 0.2 leads to higher MSE values for $\overline{E_{in}}$ and $\overline{E_{out}}$.

The most significant difference from using the weight decay regularization in the model is the dramatic decrease in over-fitting, which allows the model to generalize better and have lower values of $\overline{E_{out}}$ even at a high polynomial degree. The direct comparison of the model with and without weight decay regularization applied can be seen in Figure 3. Without regularization, a small training dataset and a high model degree polynomial of 64 causes the values of E_{out} to be above 8. Showing that it does not generalize with poor performance. For the same model with regularization applied, the E_{out}values are much smaller, under 1, showing a considerable difference and the ability for that model to better generalize.

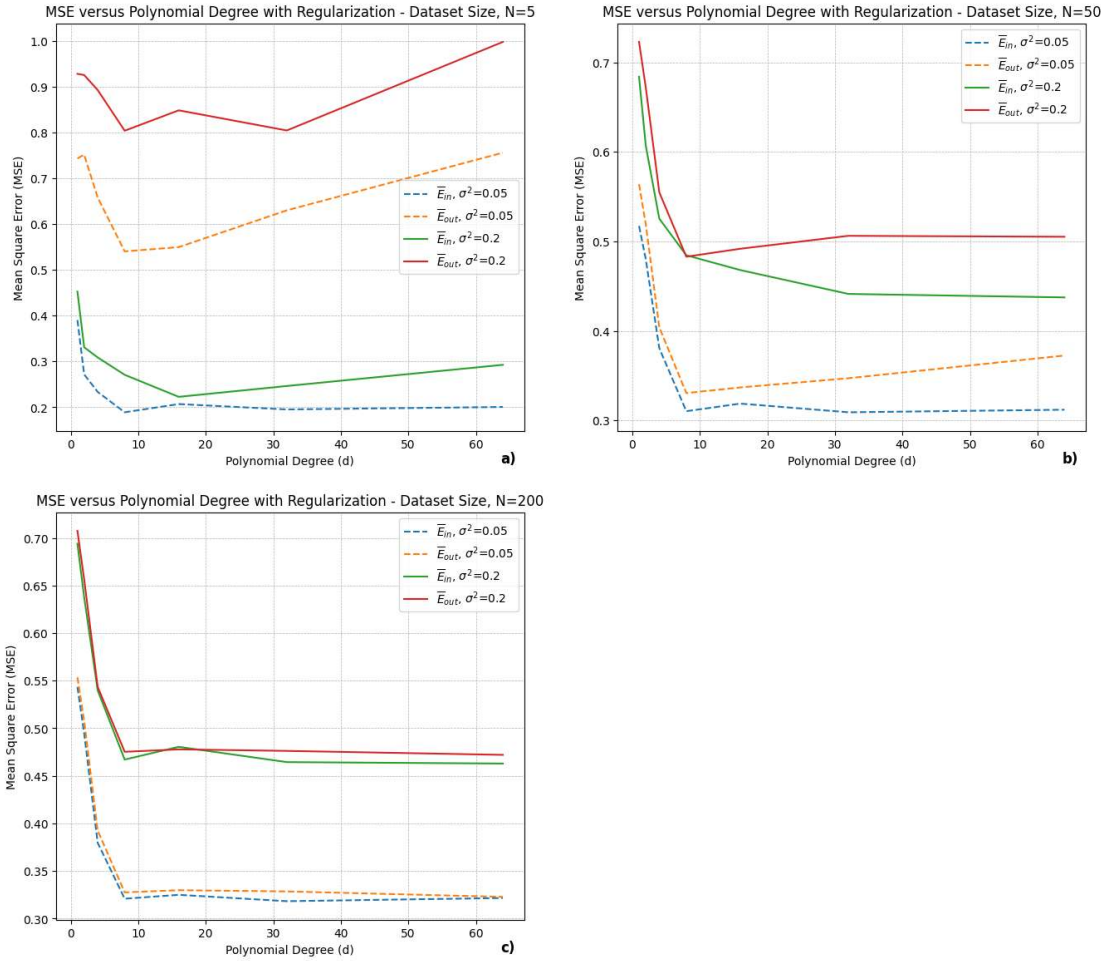

Figure 2: *Mean square error plotted against the polynomial degree of the regression model that uses weight decay for regularization. Each figure is plotted against a fixed dataset size N and plots the data for* $\overline{E_{in}}$, $\overline{E_{out}}$, *at a specified variance values.*

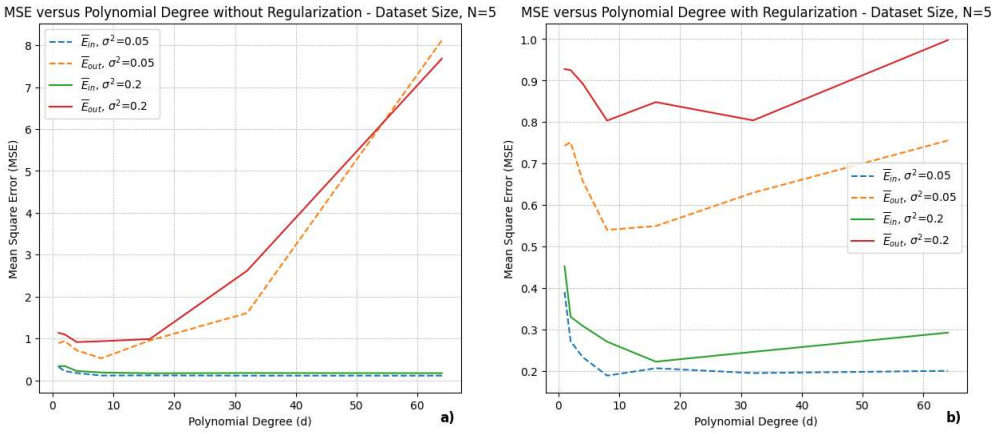

Figure 3: *Comparison of the mean square error plotted against the polynomial degree for regression models that do not use weight decay regularization (a) and a regression model that does (b). Figure a) shows a large amount of overfitting for the model without regularization when the dataset is small, but the polynomial degree is large. Figure b) shows how the regularization can help the same model generalize better and decrease the value of* $\overline{E_{out}}$.

Sample Size (Dataset Size) and Noise:

The sample size, or dataset the model is trained on, determines how well it can be generalized to data outside the training set. The graphs created use the following regression model settings, with fluctuating details represented on the graph axis, including mean square error and training dataset size.

- $N \in \{2, 5, 10, 20, 50, 100, 200\}$
- $\sigma^2 \in \{0.05, 0.2\}$
- $d \in \{2, 8, 32\}$
- $M = 50$
- $Learning\ Rate = 0.01$
- $Iterations\ of\ GD = 2000$

Results Without Regularization:

Without regularization, the results in Figure 4 show that initially $\overline{E_{in}}$ is the lowest at small training data set sizes and increases until it reaches an eventual plateau at larger training sizes. The opposite is true for $\overline{E_{out}}$ where it starts at its highest MSE value when a model is trained on a small dataset and decreases until it reaches a plateau close to the value of $\overline{E_{in}}$.

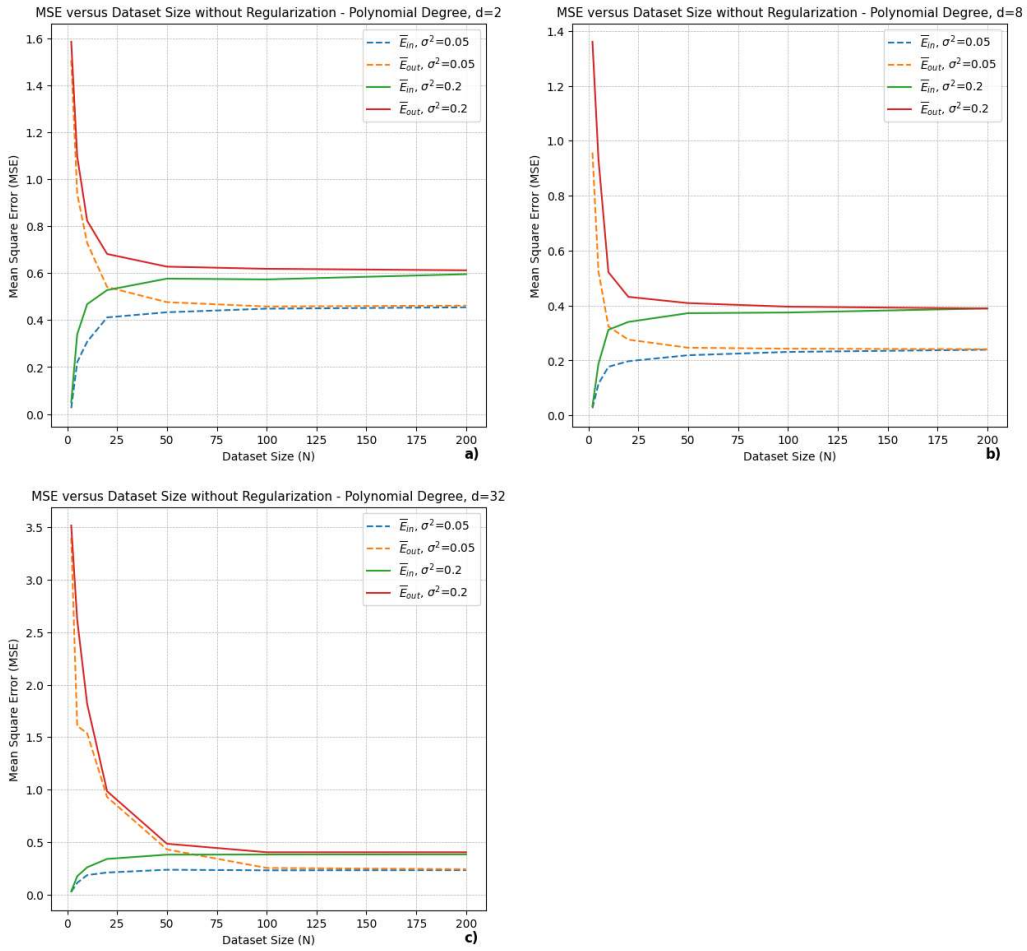

Figure 4: *Mean Square Error plotted against the dataset size of the regression model. Each figure is plotted against a fixed polynomial degree d and plots the data for* $\overline{E_{in}}$, $\overline{E_{out}}$, *at a specified variance values.*

Figure 4 above also shows the impact a function with a large polynomial size and a small training dataset can have. As an example, Figure 4a) with a polynomial size of 2, has a much smaller starting MSE for $\overline{E_{out}}$ in comparison to 4c), which has a polynomial size of 32.

Results With Weight Decay Regularization and Comparison:

Weight-decay regularization was once again implemented with the following settings to the gradient descent function and plotted:

- $Weight\ Decay\ Value = 0.1$

With weight decay regularization in place, as shown in Figure 5, all the starting values for $\overline{E_{out}}$ at small training dataset sizes, are much less than in Figure 4 without regularization. The starting $\overline{E_{in}}$ stays approximately the same after using the weight decay.

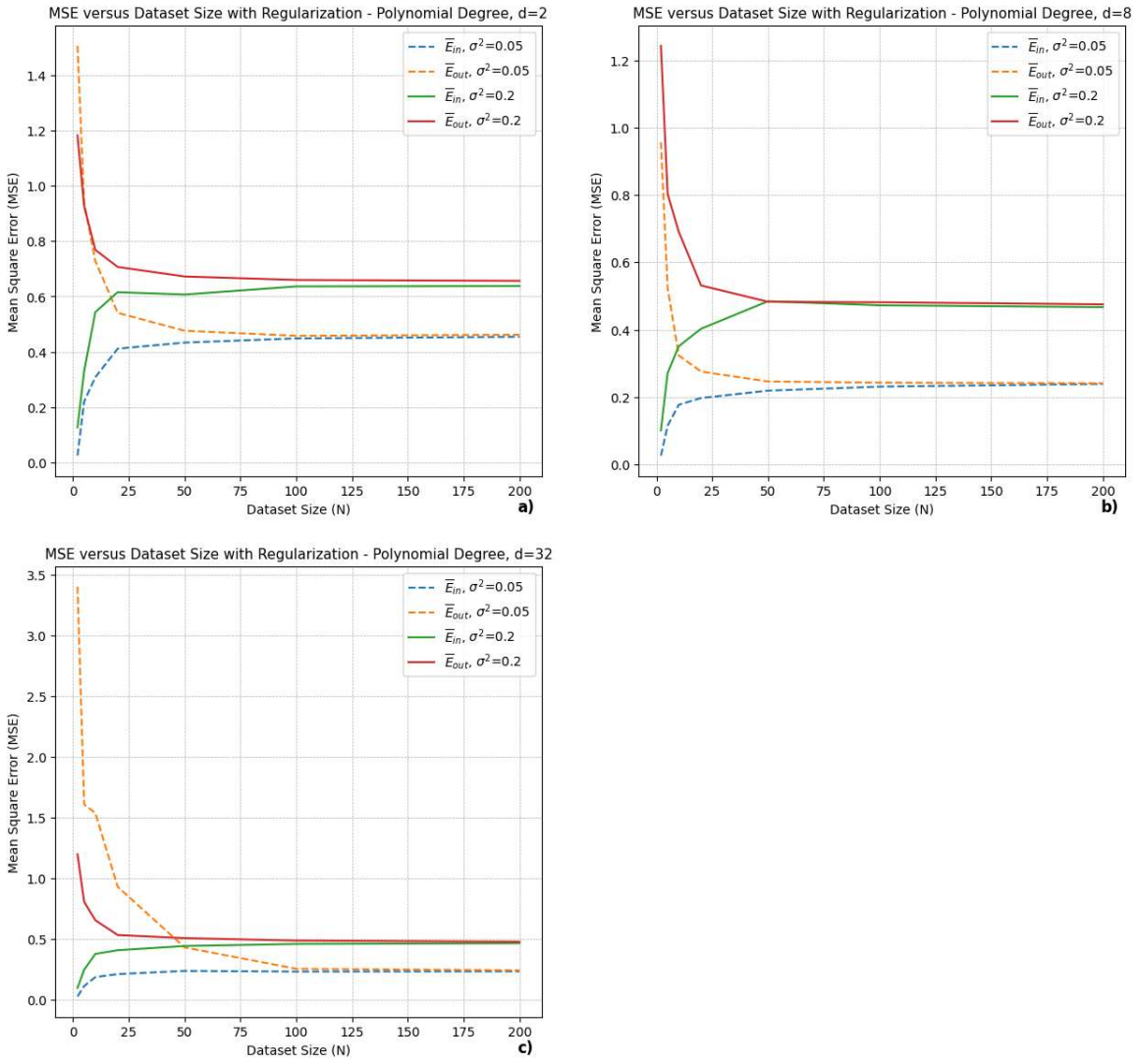

Figure 5: *Mean square error plotted against the dataset size of the regression model that uses weight decay for regularization. Each figure is plotted against a fixed polynomial degree d and plots the data for $\overline{E_{in}}$, $\overline{E_{out}}$, at a specified variance values.*

When comparing the data in Figures 6a), 6c) without regularization and to the data in Figures 6b), 6d) with regularization, respectively, it's clear the starting $\overline{E_{out}}$ is much lower when the model uses regularization. The other trend that appears, which was not evident until the graphs were side by side, is that for all fixed polynomial degrees, the MSE values of $\overline{E_{in}}$ and $\overline{E_{out}}$ converge at slightly larger values when using regularization for any variance value. This may be caused by the weight decay term changing the slopes of the convergence line to "smooth" them so they generalize better on data.

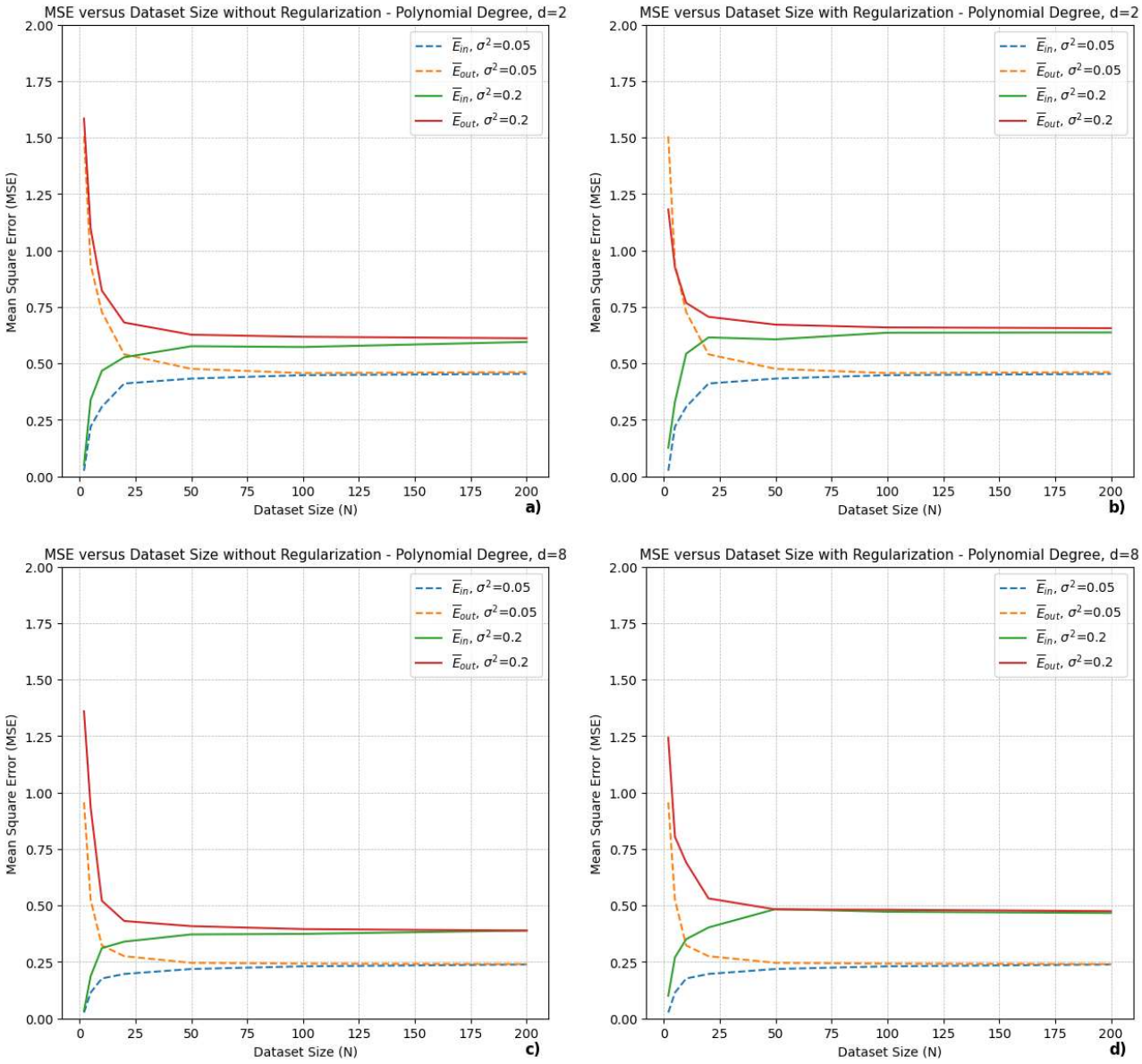

Figure 6: *Comparison of the mean square error plotted against the training dataset size for regression models that do not use weight decay regularization a), c) and a regression model that does (b), d). Figure a),c) show larger values of $\overline{E_{out}}$ for small datasets (overfitting). Figure b) and d) shows how the regularization can help the same model generalize better and decrease the value of $\overline{E_{out}}$ for small datasets.*

Conclusion on Dataset Size, Polynomial Degree, Noise:

The data shown and graphed above highlighted a few meaningful relationships. It showed that the relationship between the training sample size and the dataset is significant for finding the optimal regression/ML model parameters. A model with a high degree polynomial and minimal dataset will be overfitted, causing it to generalize poorly on other data. A model created with lots of training data but a small degree polynomial will cause it to underfit the data and generalize poorly. The relationship between under and overfitting is vital so the model can generalize to other data well.

The weight-decay regularization algorithm is beneficial to help prevent overfitting the model to the data. Figure 3 shows how the regularization prevents overfitting and drastically reduced $\overline{E_{out}}$ for a high degree polynomial on a small dataset. The regularization helps the model generalize better and smooths the slope of the $\overline{E_{in}}$, $\overline{E_{out}}$ curves.

Increased noise in the data resulted in an MSE value for the $\overline{E_{in}}$, $\overline{E_{out}}$ curves to be higher for all models. Models trained on larger datasets also performed better with higher noise values, which is expected since it will help reduce the impact of large outliers in the dataset.

Modifying Hyperparameters- Learning Rate, Weight Decay Value, Trials:

In this section, I'll discuss some of the trends observed while changing the hyperparameter values like the learning rate, weight decay and number of trials used to build $\bar{E}$. These observations were not explored as in-depth as the above sections but will be discussed.

All graphs in this section will be plotted using the regression model with polynomial degree 16. This degree seemed to perform well in all scenarios and resulted in lower MSE values for $\overline{E_{out}}$. The other settings will be as follows until the value is explicitly changed in the section to look for patterns:

- $d = 16$
- $\sigma^2 \in \{0.05, 0.2\}$, one value used per line plotted.
- $N \in \{2, 5, 10, 20, 50, 100, 200\}$one value is used and fixed in each graph.
- $M = 50$
- $Learning\ Rate = 0.01$
- $Iterations\ of\ GD = 2000$
- Weight Decay = 0.1

Learning Rate:

The learning rate appears to vary how quickly the model will approach the ideal coefficient values and minimize the converge point of $\overline{E_{in}}$ and E_{out} of the model. For a minimal learning rate, like in Figure 7a), it appears the model coefficients will underfit and get stuck, resulting in a higher value MSE value for $\overline{E_{out}}$. For a larger learning rate, like in Figure 7c), the model will quickly change to converge at the point where $\overline{E_{in}}$ and $\overline{E_{out}}$ are similar. The downside of the higher learning rate appears to be the model can converge too quickly to a suboptimal solution with a higher MSE convergence value. However, In the data shown in Figure 7, the higher learning rate of 0.2 appears to be the better solution to the problem and should have been used instead.

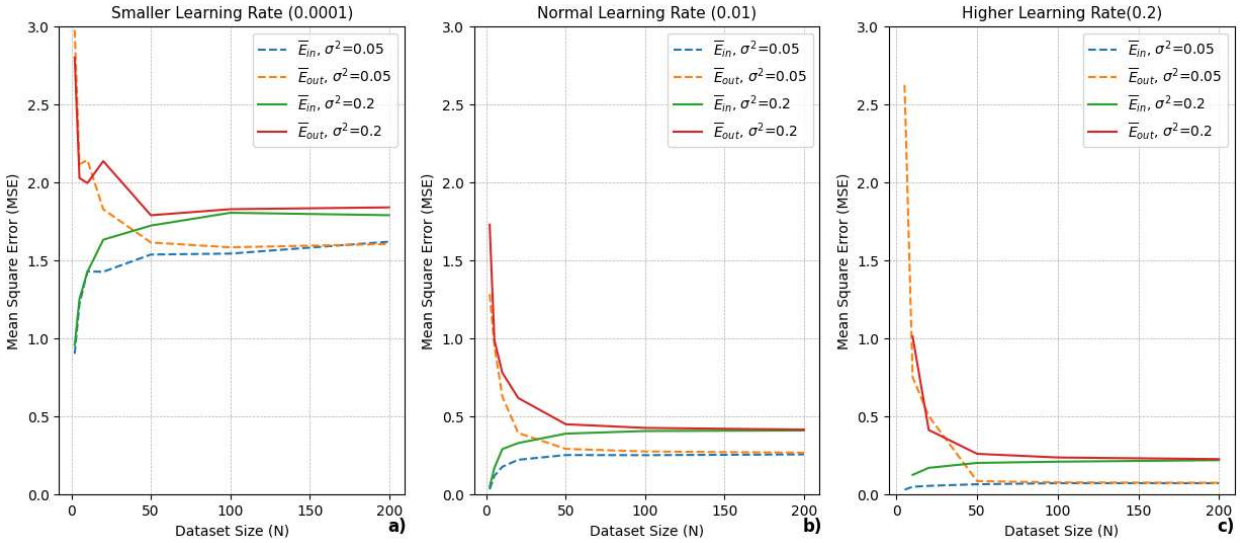

Figure 7: *Comparison of different learning rate effects on the mean square error of $\overline{E_{in}}$, $\overline{E_{out}}$, with varying dataset sizes. The lower learning rate appears to underfit the data and has a high MSE value. The higher learning rate quickly fits the data at a lower MSE value for $\overline{E_{in}}$, $\overline{E_{out}}$.*

Weight Decay:

With modified values of weight decay, we can see the generalization pattern it has on the regression model. For smaller values of weight decay, as shown in Figure 8a), the values of $\overline{E_{out}}$ are much larger for a small training dataset than in Figure 8c). The smaller values do not stop the overshoot/overfitting effect as much as the higher value weight decays. The smaller weight decay also causes the model to converge more slowly and requires more iterations for similar results.

With a high weight decay value, the $\overline{E_{out}}$ values are much smaller, with a smoother curve to the graph. The downside of large weight decay values is that the model can be hard to fit perfectly into the data.

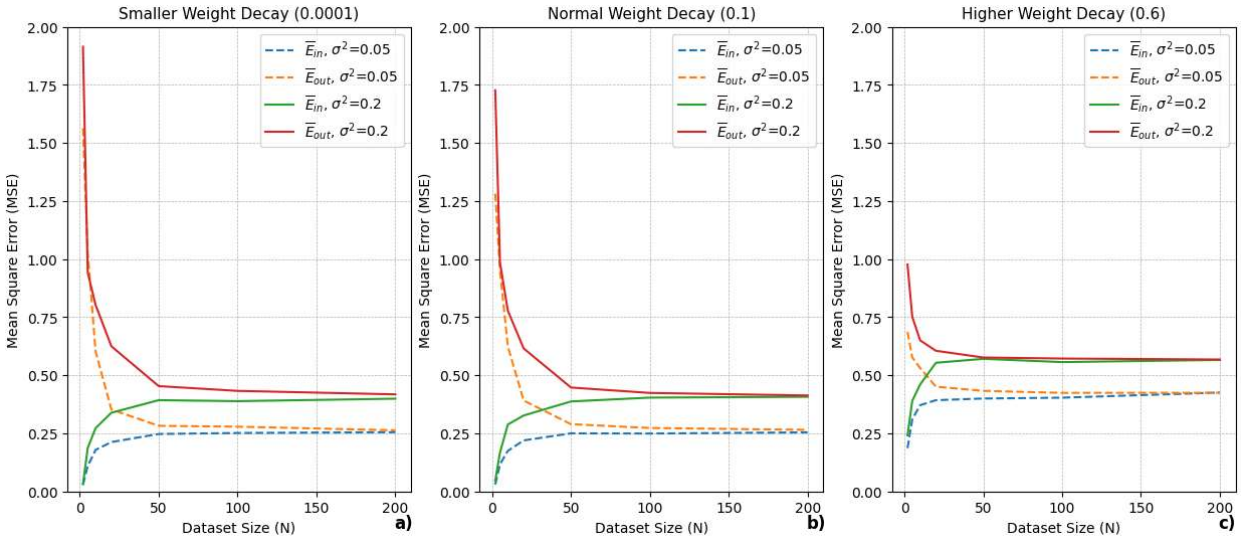

Figure 8: *Comparison of different weight decay effects on the mean square error of* $\overline{E_{in}}$, $\overline{E_{out}}$, *with varying dataset sizes. The smaller weight decay causes a slower convergence of the network.*

Number of Trials- M:

While testing multiple values of M, the number of trials used to get $\overline{E_{in}}$ and $\overline{E_{out}}$, it was apparent that if the number M was low, the graphs could be quite different between the different trial runs. This resulted in inconsistent patterns between trials because of the random nature of the data. It was sometimes difficult to see trends or consistent data shapes with M < 10. With M > 20, the data shapes and trends became more apparent and easier to analyze.