COMP 551 : Assignment 2

Group 8 : Ayaz Ciplak - 261013785, Jacqueline Silver - 261012002, Christopher Smith - 261013926

Abstract

In this assignment, we investigated the effects of varying model complexity on linear regression models, and explored regularization techniques to manage overfitting.

Our first task involved generating synthetic data from a non-linear function to which we applied Gaussian basis functions, gradually increasing the model's dimensionality. By fitting linear regression models with these bases, we explored how increasing complexity influenced the model's ability to approximate the underlying data distribution. Next, we explored the bias-variance trade-off, repeating model fits with varying amounts of Gaussian bases on multiple noisy datasets. We examined the trends in average train and test error to observe how model complexity affects bias and variance. The third task of the assignment focused on regularization. We implemented L1 (Lasso) and L2 (Ridge) regularization in our models, and evaluated the effects of varying lambda values through 10-fold cross validation. By plotting train and test errors as well as bias-variance decompositions, we identified optimal lambda values for both L1 and L2 regularization. Lastly, we studied the impact of L1 and L2 regularization on loss functions and optimization paths. Through contour plots and gradient descent trajectories, we visualized how L1 regularization induces sparsity, while L2 penalizes large weights without eliminating features.

Overall, this assignment increased our understanding of balancing model complexity, controlling overfitting, and optimizing model performance through regularization. Some important findings included the difference in behaviors between L1 (sparsity-inducing) and L2 regularization, as well as how noise influences bias-variance tradeoff plots.

Task 1: Linear Regression with Non-Linear Basis Functions

In task 1, we explored how increasing model complexity using Gaussian basis functions affects the performance of linear regression on our dataset.

Data Generation: We generated 100 datapoints sampled from the following non-linear function, and plotted our data as per *Figure 1*.

$$y(x) = \sin(\sqrt(x)) + \cos(x) + \sin(x) + \epsilon$$

Gaussian Basis Functions: We generated plots with varying amounts of Gaussian basis functions, ranging from 0 to 110 in increments of 10. These were generated from the following function (with the standard deviation value set to 1), as per *Figure 2*.

$$\phi(x, \mu, \sigma) = exp(-\frac{(x-\mu)^2}{\sigma^2})$$

Model Fitting: A model was trained on each amount of Gaussian bases D (i.e. D = 0, 10, ..., 110). This illustrated how model complexity changes based on the number of basis functions used, and can be viewed in *Figure 3*, *Figure 4* and *Figure 5*.

Model Selection: We split the data into a training and validation set, and computed the SSE as per the function below. The numerical results can be seen in *Figure 6*. We selected an optimal model based on the one that minimizes validation set error (SSE), i.e. generalizes best to unseen data and reaches an equilibrium between overfitting and underfitting. For our dataset, this was the model using <u>20 Gaussian bases</u>.

Explanation:

- **Model Performance - Observations:** For models with lower amounts of Gaussian bases (0-50), we observe predictable trends in the model performance: As the number of bases increases, the model can more closely fit the training data. We therefore see consistently decreasing MSE values on the train set. Additionally, as the number of bases increases, we run a higher risk of overfitting the training data. We can therefore see that although the test SSE diminishes consistently for 0, 10, 20 bases (model approaches actual distribution), it starts climbing again for 30, 40, etc. bases (model overfits training data). Finally, we observe an equilibrium between underfitting and overfitting at the point where the test error is minimized.

  For models with higher amounts of Gaussian bases (60+), we observe the effects of overfitting and numerical instability. The latter is caused by having a large number of closely-spaced Gaussian transformations with narrow spreads (sigma = 1). This can result in the design matrix becoming ill-conditioned (columns of

matrix are highly correlated), leading to a matrix that is very difficult to invert. This ultimately leads to extremely large Gaussian bases being used and the model becoming unpredictable, which explains the SSE values exploding after $D > 50$.

- **Optimal Model Selection - Reasoning:** The validation set helps us measure how each given model will perform on unseen data. Since the goal of our model is to make accurate predictions on new data, this is a metric we want to prioritize when selecting an optimal model. Our optimal model will therefore be the one that minimizes validation set error, i.e. generalizes best to unseen data and reaches an equilibrium between overfitting and underfitting.

Task 2: Bias-Variance Tradeoff with Multiple Fits

In task 2, we sought to explore the tradeoff between bias and variance as we shift from a model that underfits the sample data to one that overfits.

To achieve this, we started by repeating the data generation process seen in Task 1 ten times (i.e. generating ten synthetic datasets, each consisting of 100 data points, from the same non-linear function with noise). We then proceeded to fit a Linear Regression model to each of the datasets, gradually increasing model complexity by applying an increasing number of Gaussian bases (ranging from $D = 0$ to $D = 100$).

The results seen in *Figure 7*, *Figure 8* and *Figure 9* show that when model complexity is low (for $D = 0$ to $D = 10$), the model tends to underfit, resulting in high bias and low variance. This comes as no surprise given that we are fitting a linear model to non-linear data. Conversely, when model complexity is high (for $D \geq 50$), the model overfits the data, leading to low bias but high variance. Visually, we can conclude that somewhere between $D = 20$ and $D = 40$ there is an optimal level of complexity which results in a balance between underfitting and overfitting, and therefore balance between bias and variance.

To quantifiably measure the effects of increasing model complexity on model performance we split the data into training and test sets. In *Figure 10*, we observe a steady decline in the average MSE of the training set, as the number of Gaussian bases (D) increases from 0 to 50, as the model begins to better fit the data and bias decreases. However, at $D = 50$, we notice a sharp increase in the average training error as the model begins to overfit, leading to an increased variance. As for the test set, the average MSE also follows this initial trend, with a decrease in average test MSE from $D = 0$ to $D = 10$ as the model starts to fit better to the non-linear data and the bias decreases. However, unlike the training set, from $D = 20$, the average test error exhibits a gradual rise as the model begins to overfit the data. This is then followed by a sharp increase at $D = 50$ due to extreme model overfitting and therefore a high variance.

We can therefore conclude that between $D = 10$ and $D = 30$ the model transitions from underfitting to overfitting, and therefore, in this range lies an optimal level of model complexity that balances the tradeoff between bias and variance.

Task 3: Regularization with Cross-Validation

Implementing Regularization: We modified our existing linear regression model to include L1 and L2 regularization. This was done by extending our basic model with Gradient Descent optimization, then implementing regularization within it. Our model takes in two additional lambda values: one for L1 and one for L2 regularization. To use one type of regularization, we simply set the other lambda value to 0.

Cross-Validation: To implement 10-fold cross-validation in this step, we first implemented a function to shuffle and split our data, such that the validation set at each iteration contains a completely randomized subset of our data. For each type of regularization, we used 8 lambda values, ranging from 0 to 0.1 for L1 regularization and from 0 to 1 for L2 regularization (as we observed that L1 regularization is more sensitive to larger lambda values). The results can be seen in *Figure 11*.

Plotting Train and Test Validation Errors: For each regularization method, we trained models using a learning rate of 0.01, maximum iterations of 10e3, 70 Gaussian bases, 10-fold cross-validation, and full-batch gradient descent. For both types of regularization, we used lambda values generated by np.logspace(-4, 2, 15). The plots can be seen in *Figure 12*.

- **Behaviour Analysis - L1 Regularization:** Since L1 regularization performs feature selection (i.e. diminishes some feature weights all the way to zero), we can observe an interesting behaviour in the MSE trends. We see large jumps in both training and test error around $\lambda = 10^{-1}$ and $\lambda = 10^{2}$, presumably at the points where the model sets a new feature's weight to zero.

- **Observations:** L1 regularization performs better at relatively lower lambda values as compared to L1. We reach a minimum in L1 test MSE at around $\lambda = 10^{-2.2}$, whereas L2 test MSE reaches a minimum at around $\lambda = 10^{-1.5}$. Additionally, the train/test MSEs for L2 regularization behave more predictably and consistently relative to a change in lambda. This is due to the fact that L2 regularization does not perform feature selection and instead scales feature weights down proportionally.

Plotting Bias-Variance Decomposition: To generate the plots for Bias-Variance Decomposition, we used the same model settings as above, and sampled from 50 different datasets each containing 40 evenly-spaced points in the range [0, 20]. However, instead of using lambda values generated by np.logspace(-4, 2, 15) for L1 regularization, we used np.logspace(-5, 1, 15) given that it is more sensitive to higher lambda values and a lower range could more clearly illustrate the trends. The bias-variance decomposition plots generated (using a logarithmic x (lambda) scale) are visible in *Figure 13* and *Figure 14*.

Selecting the optimal Lambda value:

- **Optimal Lambda:** For each type of regularization, the optimal lambda is defined as the one that minimizes the test MSE, i.e. the lambda value that produces a model that performs best on unseen data (and is closest to the actual underlying distribution). To find these optimal lambda values, we can refer to the train and test error vs. lambda graphs produced in part 3.3. For L1 regularization, our graph reaches a minimal value for test MSE (optimal lambda) at <u>lambda = 5.17947468e-03</u>. For L2 regularization, our graph reaches a minimal test MSE value (optimal lambda) at <u>lambda = 1.38949549e-02</u>.

- **Discussion - Regularization vs. Bias/Variance:** Our graphs plotting the Bias-Variance decomposition as a function of regularization values provide many insights on the effect of regularization (L1 and L2) on a model's ability to generalize.

  - **Bias-Squared Trend:** Bias refers to the difference between the model's predicted output and the true output on average. As regularization increases (higher lambda values), the bias-squared value goes up. This is because the model gets increasingly simple as we increase regularization, and deviates more and more from the true underlying distribution.
  - **Variance Trend:** Variance measures the sensitivity of the model to small fluctuations in the training data. A higher variance value can mean overfitting, since it indicates that a model's predictions change significantly when trained on different datasets. We observe this trend in the graphs, since as regularization increases, the variance value lowers. This indicates that as we increase regularization, the model overfits less and less, and it adjusts less and less to small fluctuations in the training data (performs more similarly on unseen data).
  - **Bias Squared + Variance vs. Test Error:** Notice that the (bias squared + variance) graph is the exact same as the test error graph minus a value of 1 (i.e. the variance of the noise that was added to each data sample). This is a good example of Bias-Variance Decomposition, which cites that Test Error = $Bias^2$ + Variance + (Noise Error).
  - **Comparison of Bias-Variance Decomposition Graphs:** Although the Bias-Variance decomposition graphs for L1 and L2 regularization behave similarly, there are some minor differences between them. One example is the fact that the trends in L1 regularization seem to be more "compressed" than those in L2 regularization. This is due to the fact that L1 regularization is more sensitive to fluctuations in lambda values, since higher regularization values can completely exclude some features (a phenomenon known as feature selection).

Task 4: Effect of L1 and L2 Regularization on Loss

In task 4, we looked to investigate and compare the impact of Lasso (L1) and Ridge (L2) regularization on our linear regression model. Specifically, we analyzed these techniques' effects on the model's loss function and visualized the change in the its gradient descent path as we varied the regularization strength (λ).

Data Generation: In order to conduct this investigation, we generated another set of synthetic data consisting of 50 data points that are uniformly distributed between 0 and 10, gathered from the following linear function with

Gaussian noise: $y = -4x + 10 - 2\epsilon$, where ϵ is Gaussian noise with a mean of 0 and variance of 1 (visible in Figure 15).

Applying L1 and L2 Regularization: With the dataset generated, we applied L1 and L2 regularization onto our linear regression model using various values of λ. We observe the following results, seen in *Figure 16*, *Figure 17*, *Figure 18* and *Figure 19*:

- **Lasso Regularization (L1):** At lower levels of regularization strength (λ), the linear fit is relatively close to the true underlying function. As λ increases, higher weights are penalized. The weights shrink, resulting in an increasingly sparse model. As a result, the predicted fit deviates from the underlying function. The model essentially becomes less flexible with lower coefficients (and less parameters overall) causing the model to underfit the data.

- **Ridge Regularization (L2):** Similarly to L1, at lower levels of λ, the linear fit aligns almost perfectly with the underlying function, which is potentially a sign of overfitting. As λ increases, L2 regularization penalizes the sum of squared weights, thereby penalizing larger weights, which in turn causes the fit to pivot clockwise again. However, unlike L1 regularization, L2 does not promote sparsity in the model, and so the weights are diminished more smoothly, resulting in a flatter slope, and slightly less significant underfitting.

Effects of L1 & L2 on Loss Function: We first decided to visualize the model's loss function and each regularization technique's penalty (see *Figure 20*) before adding them together to analyze the effects of regularization on the loss function (as illustrated by *Figure 21*) :

- **Loss Functions with L1 Regularization:** Without Regularization ($\lambda = 0$), the cost function contours are linear, and almost parallel lines, corresponding to the pure cost function (without penalties) for a linear relationship. As λ increases, the contours of the cost function begin to form more complex shapes. In the case of L1 Regularization, the contours take on a diamond-like shape. This is because L1 regularization tends to force weights toward zero, which is reflected in the sharper corners of the contours. The sharpness around the origin means that gradient descent is more likely to drive some weights exactly to zero, promoting sparsity in the model (i.e. some features are dropped entirely from the model).

- **Cost Functions with L2 Regularization:** Without Regularization ($\lambda = 0$), the cost function is the same. As λ increases, the L2 penalty creates contours that are increasingly circular, smooth, and closer together. This reflects how L2 regularization penalizes large weights but does not drive them to zero as aggressively as L1.

Effects of L1 & L2 on Gradient Descent (as illustrated by Figure 22):

- **Gradient Descent with L1 Regularization:** Without Regularization ($\lambda = 0$), the path follows a smooth, direct curve toward the minimum without any constraint on the weights. As λ increases, L1 pushes the model's weights toward zero more aggressively, causing the path to align with one of the axes. This reflects L1's promotion of sparsity, which encourages feature selection, which in turn results in faster convergence and reduced overfitting. Compared to L2, L1 forces one of the weights (in this case, w_0) to shrink faster and can even set it equal to zero, encouraging model sparsity.

- **Gradient Descent with L2 Regularization:** Without Regularization ($lambda$=0), the path is smooth, unconstrained, and similar to L1, moving directly toward the minimum. As λ increases, L2 shrinks weights more uniformly, resulting in smoother, circular paths, as all weights are gradually reduced without being forced to zero, which improves convergence as the isocontours become more convex. Compared to L1, L2 does not zero out weights but instead reduces their magnitude evenly, leading to smoother convergence paths without axis alignment.

Overall Effects of L1 & L2 Regularization:

- **How L1 regularization encourages sparsity (setting some weights to zero):** We have seen in the previous subsections how L1 regularization effectively works by adding a penalty ($||w||_1$) to the Linear Regression's loss function. This penalty encourages the model to drive some weights to exactly zero effectively performing a feature selection. In doing so, L1 creates a sparser model, thereby reducing model complexity and prevents overfitting. As a result, we observe improved convergence as the optimizer handles fewer non-zero parameters, focusing only on the most significant features.

- **How L2 regularization penalizes large weights but does not promote sparsity like L1:** On the other hand, L2 regularization penalizes large weights by adding a ($||w||_2^2$) term to the loss function. This encourages

all weights to shrink toward zero, but unlike L1, no weights are set to exactly zero. Instead, L2 applies a smoother, more uniform reduction in the weights. This results in a model that still prevents overfitting by limiting the effect of large coefficients, but retains all the model's features. As a result, L2 tends to provide a more stable descent path during gradient descent, reducing oscillations and ensuring a smoother convergence.

- **How different values of λ affect the optimization paths and loss landscapes:** The regularization strength (λ) controls how harshly each weight is penalized. For lower values of λ, the regularization has a lower effect, allowing the model to fit the training data more closely but risking overfitting. As λ increases, both L1 and L2 regularization alter the loss landscape, with L1 leading to sharper, square-like contours (reflecting its promotion of sparsity) and L2 creating smoother, circular contours that uniformly shrink the weights. L1's sharp contours make gradient descent paths more direct, while L2's smooth contours improve convexity leading to smoother convergence paths without axis alignment.

Creative Additions

In order to thoroughly analyze the effects of regularization, we extended our investigation to explore the joint application of L1 (Lasso) and L2 (Ridge), known as "Elastic Net Regularization".

In *Figures 23* and *24*, we observe the linear fit of our model begin to gradually underfit the data as we increase the value of λ, similarly to what was seen in task 4. However, we note that with this elastic net regularization (or joint application of L1 and L2 regularization), the effects of regularization happen at a faster pace.

We then plotted both the L1 regularization and L2 regularization penalties, and compared it to the sum of the two penalties, which creates the Elastic Net regularization penalty contours, as seen in *Figure 25*. We observe that the sum of the two penalties creates an almost diamond-like shape in the center, reflecting a balanced mix of L1 and L2.

We can discern the effects of each type of regularization in *Figure 26* as we vary the value of regularization strengths, λ_1 and λ_2, initially with values where $\lambda_1 > \lambda_2$, then where $\lambda_1 < \lambda_2$, and finally $\lambda_1 = \lambda_2$. In this first case, as expected, where L1 regularization is more dominant, we note sharper edged contour lines. When L2 dominates, however, we notice smoother, more circular, and closer together contours, although less so than when it was just L2 regularization. And finally, when $\lambda_1 = \lambda_2$, we have a balanced mix of the benefits of L1 and L2 regularization, creating a round contour lines, where the innermost contour is relatively more sharp-edged.

In *Figure 27*, we observe how the gradient descent path is shortened as the regularization strengths are increased, regardless of whether that is L1 or L2, as L1 promotes sparsity and L2 encourages smoother convergence due to its convex contours.

Ultimately, Elastic Net regularization (the joint application of L1 & L2 regularization) combines the strengths of both L1 (sparsity) and L2 (smooth optimization), offering a balanced approach of effectively controlling overfitting in our models.

Discussion & Conclusion

In this assignment, there were many key takeaways from our exploration of how model complexity and regularization influences the performance of linear regression models. This allowed us to gain major insight into how these aspects impact model behavior and optimization.

By experimenting with non-linear basis transformations on our linear regression model, we observed the balance between underfitting and overfitting, and by association, the tradeoff between bias and variance. As a result, we now understand the importance of finetuning model's level of complexity in order to have an effective model.

We also compared the effects of Lasso (L1) and Ridge (L2) regularization on improving model generalization. We experimented with different values of regularization strength λ for each type of regularization, and concluded how these different aspects influence the model's loss function, gradient descent, and overall behavior. We noted the significant impact of λ on the model's performance, and the distinction between L1 and L2 regularizations behaviors, specifically regarding L1's tendency to encourage sparsity and L2's smoother optimization.

Further exploration could involve various extensions of this work. For instance, our experiment with applying L1 and L2 regularization together, to form Elastic Net regularization. Additionally, we could look into incorporating our regularization techniques into our logistic regression model seen in the previous assignment; or even how different non-linear basis functions, such as Sigmoid or polynomial bases, might play into increasing model complexity.

Overall, the tasks within this assignment allowed us to learn extensively about the large roles that model complexity and regularization play in machine learning models, while also providing a strong foundation for further exploration topics.

Contributions:

The project was a collaborative effort between Ayaz, Christopher, and Jacqueline. Specifically, Task 1 was focused on by Ayaz and Jacqueline, Task 2 was focused on by Christopher, Task 3 was focused on by Ayaz, Task 4 was focused on by Christopher, and the report was focused on by all team members.

Appendix:

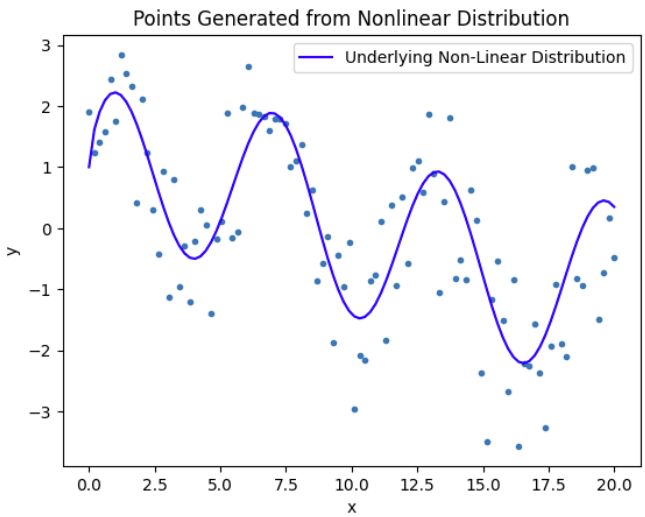

Figure 1: Generated Data Points - Task 1

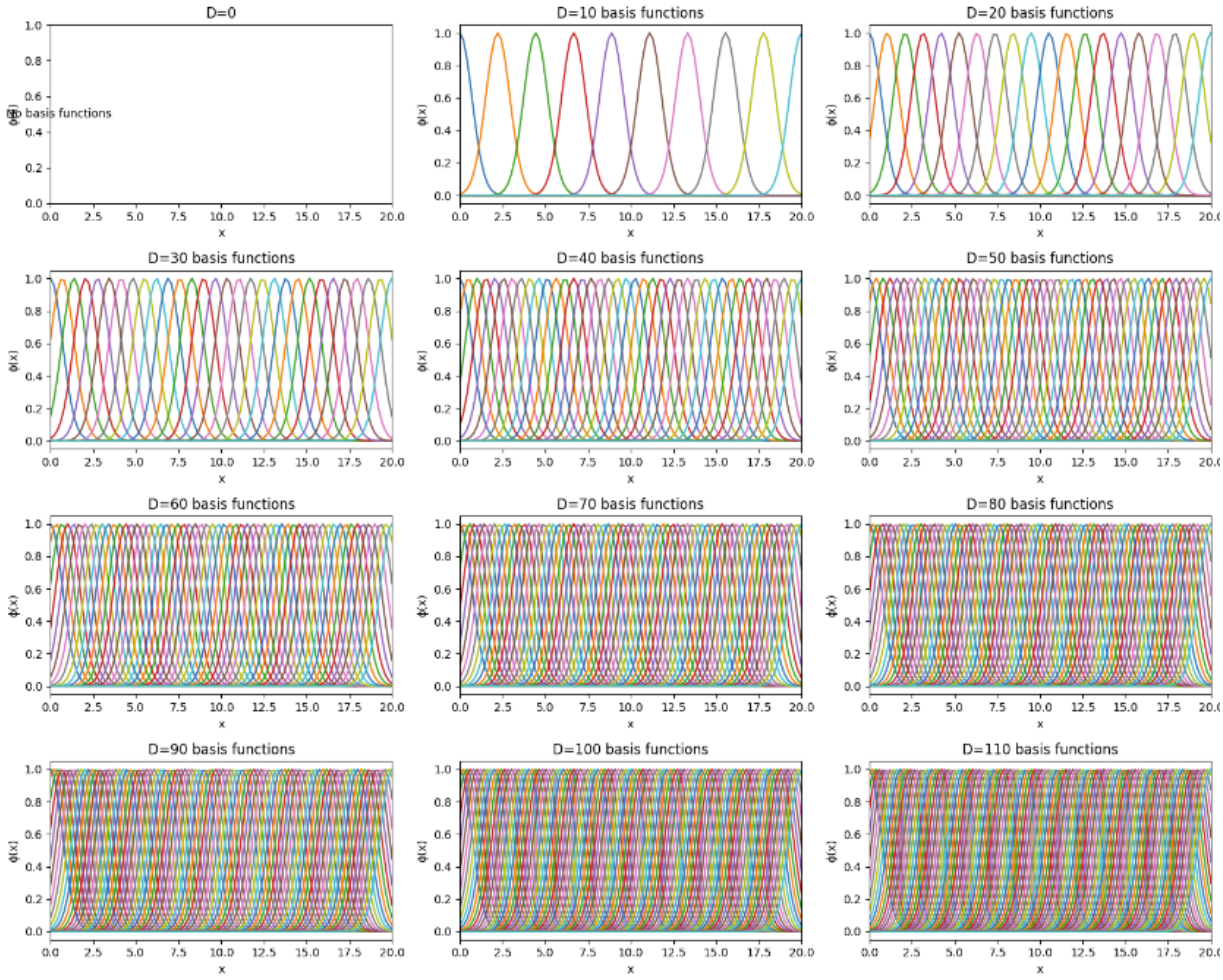

Figure 2: Plots of Varying Amounts of Gaussian Bases - Task 1

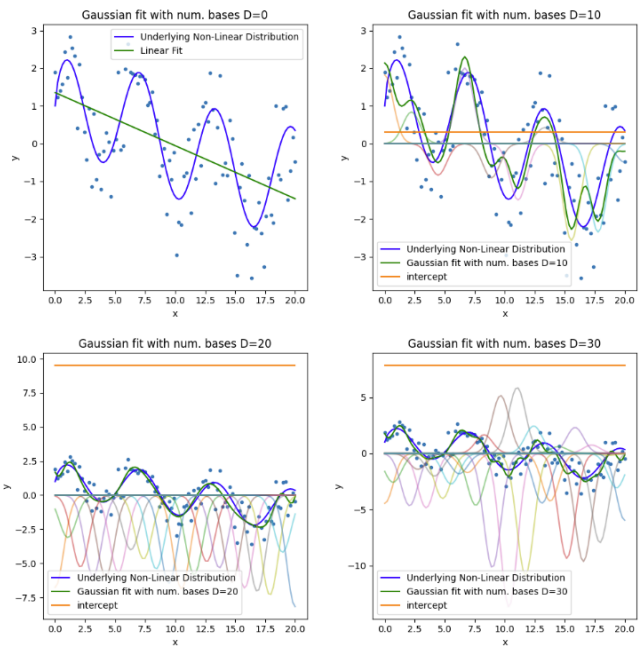

Figure 3: Models Generated with Varying Gaussian Bases (0-30) - Task 1

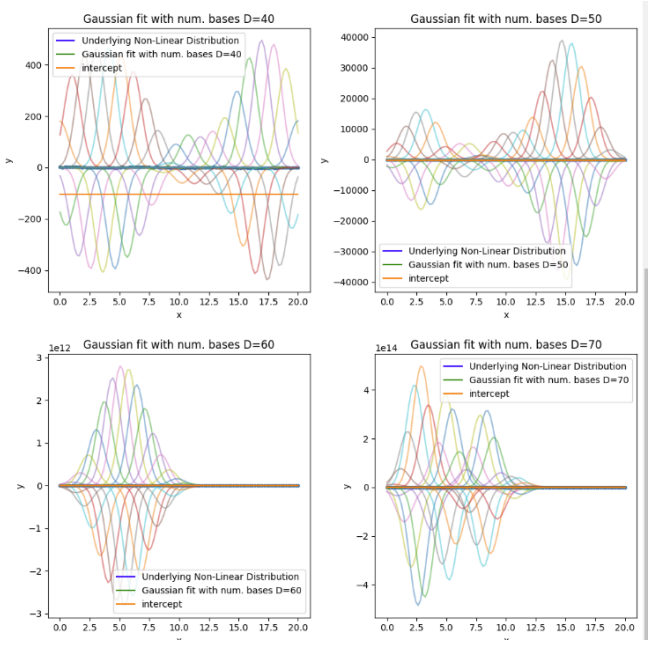

Figure 4: Models Generated with Varying Gaussian Bases (40-70) - Task 1

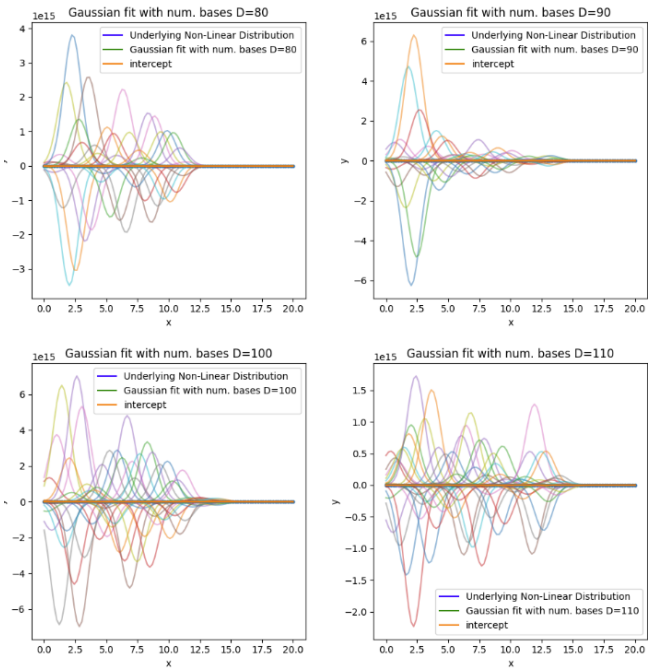

Figure 5: Models Generated with Varying Gaussian Bases (80-110) - Task 1

SSE Metrics for Models with Varying Numbers of Gaussian Bases

Number of Bases	Train SSE	Test SSE	Train MSE	Test MSE
0	64.85001952409635	16.181908986612353	0.8106252440512044	0.8090954493306176
10	32.76333699415648	18.515071905092558	0.40954171242695603	0.9257535952546279
20	21.256367854857157	14.00174096864689	0.26570459818571446	0.7000870484323445
30	18.803762750817988	15.714901296150328	0.23504703438522484	0.7857450648075164
40	16.34704741114147	16.848501677922425	0.2043380926392684	0.8424250838961213
50	11.266000517740927	42.03865012783115	0.1408250064717616	2.1019325063915573
60	96161402132.78903	81611493750.52188	1202017526.659863	4080574687.526094
70	4020725485 26364.75	2152352197692389.0	50259068 56579.56	1076176098846 19.45
80	52739855113413.414	70490573408167.89	659248188917.6677	3524528670408.3945
90	1214219752653065.5	7308744972565220.0	15177746908163.318	365437248628261.0
100	1421325842642259.2	1.1003818263791272e+16	17766573033028.242	550190913189563.6
110	2.1647168631555066e+17	1.631811311192982e+17	2705896078944383.0	8159056555964910.0

Figure 6: Variation in SSE as Gaussian Bases Increase - Task 1

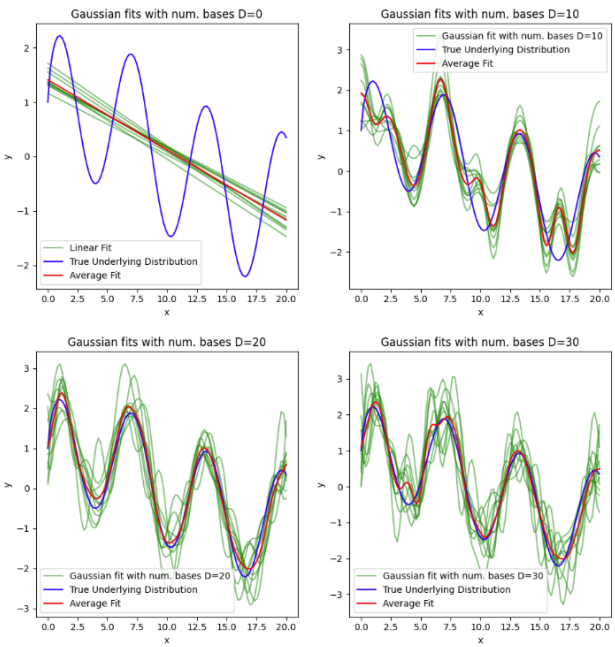

Figure 7: Models Generated with Varying Gaussian Bases (D = 0 to 30) - Task 2

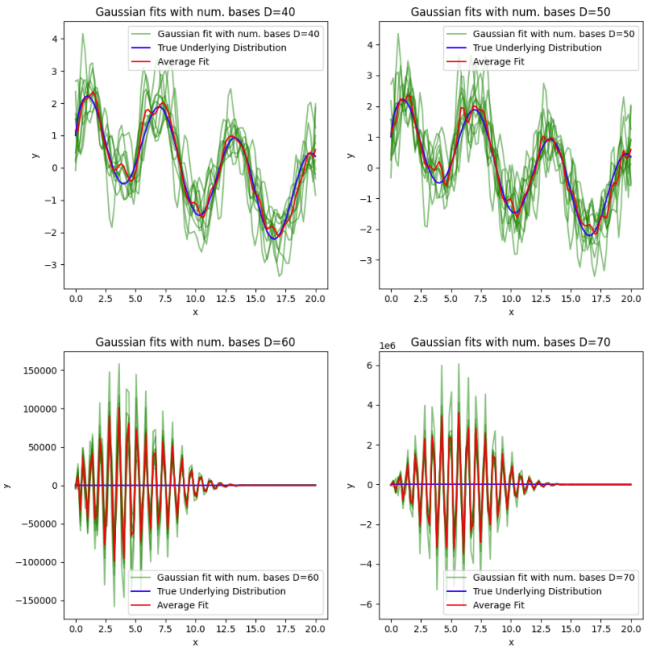

Figure 8: Models Generated with Varying Gaussian Bases (D = 40 to 70) - Task 2

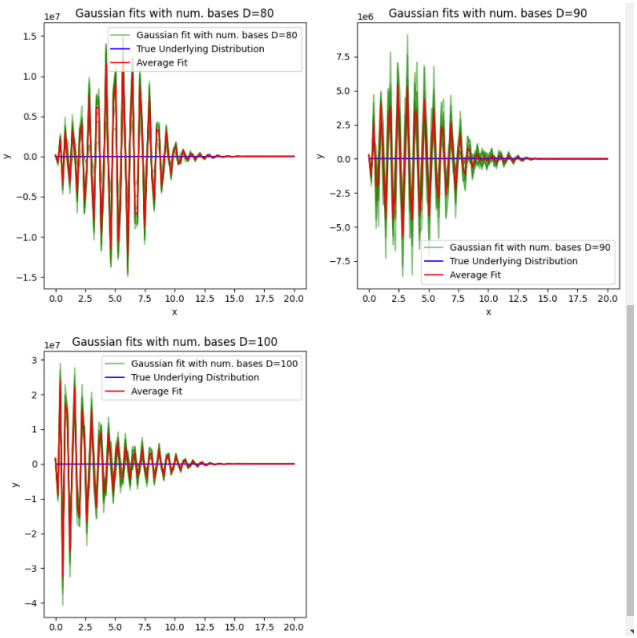

Figure 9: Models Generated with Varying Gaussian Bases (D = 80 to 100) - Task 2

Num. of Gaussian Bases (D)	Avg. Train MSE	Avg. Test MSE
0	0.94219	0.86594
10	0.54259	0.57759
20	0.33516	0.63924
30	0.27453	0.73306
40	0.22664	1.6042
50	0.17745	5.8663
60	4.2313e+10	1.1627e+15
70	2.4585e+12	3.5897e+16
80	8.299e+14	3.0177e+18
90	1.7856e+15	3.2031e+21
100	1.6972e+15	9.2263e+20

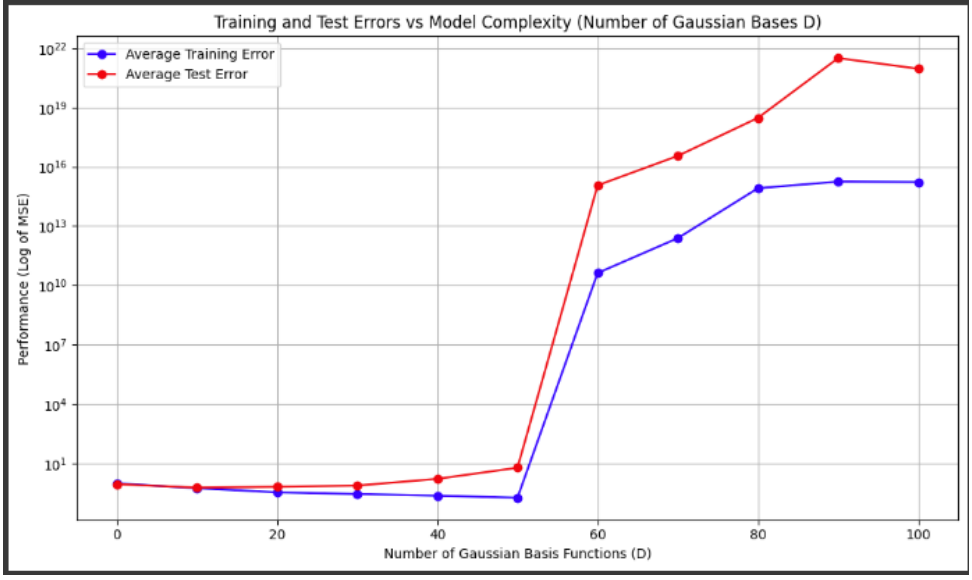

Figure 10: MSE vs. Number of Gaussian Basis Functions - Task 2

```
### L1 Regularization Stats: ###

For L1 regularization value 0 , computed avg. train MSE of 0.5523958926474435 and average test MSE of 0.8548909323659142
For L1 regularization value 0.0001 , computed avg. train MSE of 0.5526379672851476 and average test MSE of 0.8531046140321893
For L1 regularization value 0.001 , computed avg. train MSE of 0.5563045982506514 and average test MSE of 0.8369830304728186
For L1 regularization value 0.005 , computed avg. train MSE of 0.5851169798941038 and average test MSE of 0.8271648856512573
For L1 regularization value 0.01 , computed avg. train MSE of 0.6409452283682526 and average test MSE of 0.8445421156324407
For L1 regularization value 0.025 , computed avg. train MSE of 0.8833887920983668 and average test MSE of 1.0700890051790473
For L1 regularization value 0.05 , computed avg. train MSE of 1.4556708303925396 and average test MSE of 1.6677449502435315
For L1 regularization value 0.1 , computed avg. train MSE of 2.3051200944917207 and average test MSE of 2.322018390335252


### L2 Regularization Stats: ###

For L2 regularization value 0 , computed avg. train MSE of 0.5523958926474435 and average test MSE of 0.8548909323659142
For L2 regularization value 0.001 , computed avg. train MSE of 0.5537162445755378 and average test MSE of 0.850521054004424
For L2 regularization value 0.01 , computed avg. train MSE of 0.5689201434245252 and average test MSE of 0.8256229798468185
For L2 regularization value 0.05 , computed avg. train MSE of 0.6686832180239298 and average test MSE of 0.8534490156942567
For L2 regularization value 0.1 , computed avg. train MSE of 0.8126910969024566 and average test MSE of 0.9707543428648693
For L2 regularization value 0.25 , computed avg. train MSE of 1.168863821700425 and average test MSE of 1.2970644984820041
For L2 regularization value 0.5 , computed avg. train MSE of 1.5105026102640013 and average test MSE of 1.610850506174654
For L2 regularization value 1 , computed avg. train MSE of 1.816435938530482 and average test MSE of 1.8857742332146668
```

Figure 11: MSE Values for Varying Lambda Values - Task 3

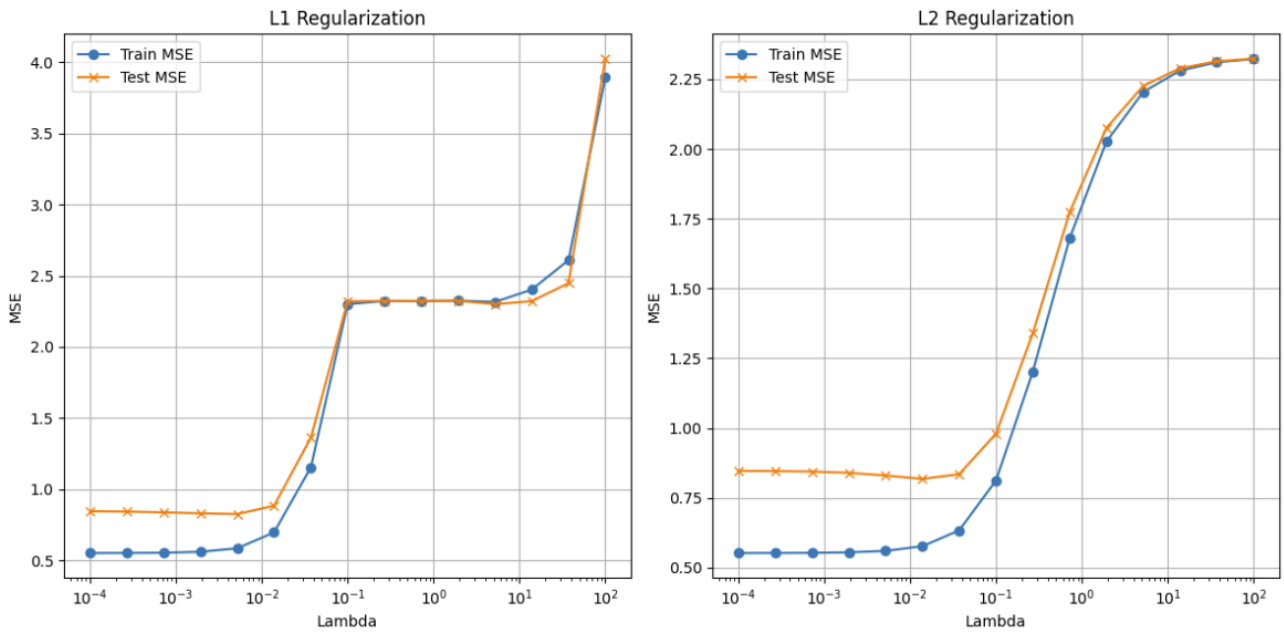

Figure 12: Train and Test MSE Plots against Regularization - Task 3

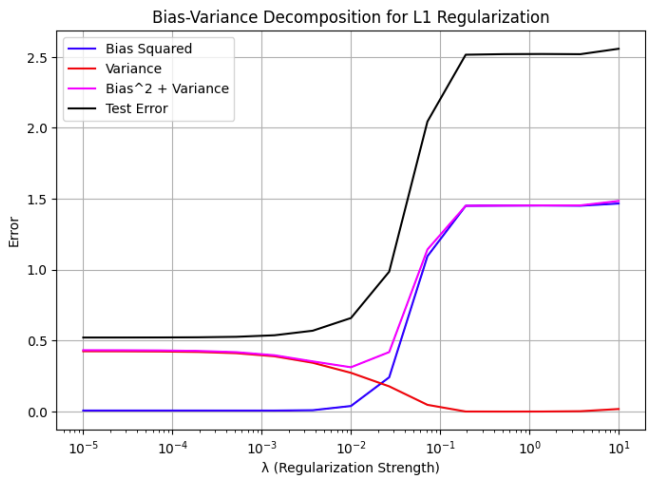

Figure 13: Bias-Variance Decomposition for L1 Regularization - Task 3

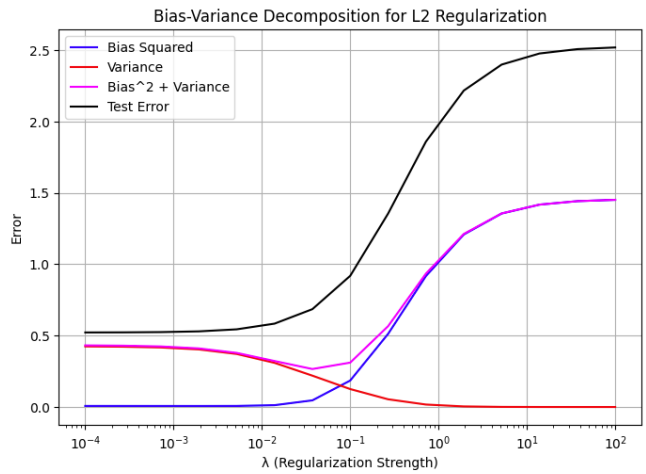

Figure 14: Bias-Variance Decomposition for L2 Regularization - Task 3

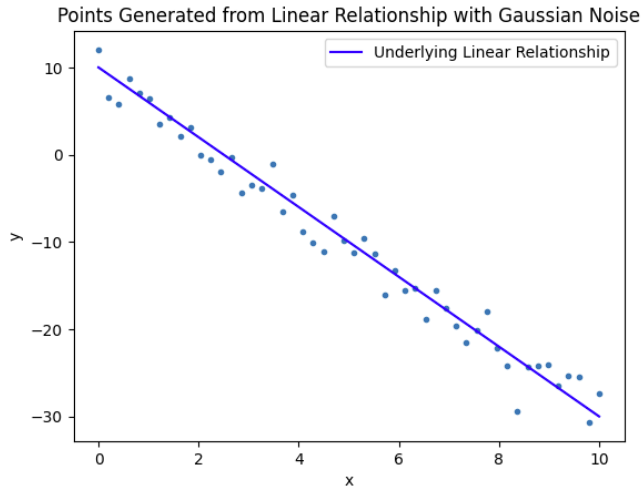

Figure 15: Points Generated from Linear Relationship with Gaussian Noise - Task 4

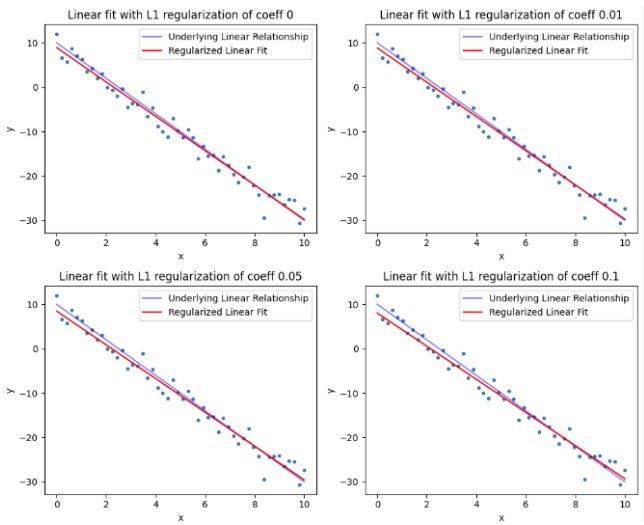

Figure 16: Linear Fit with Varying Lambdas (L1, lambda = 0 - 0.1) - Task 4

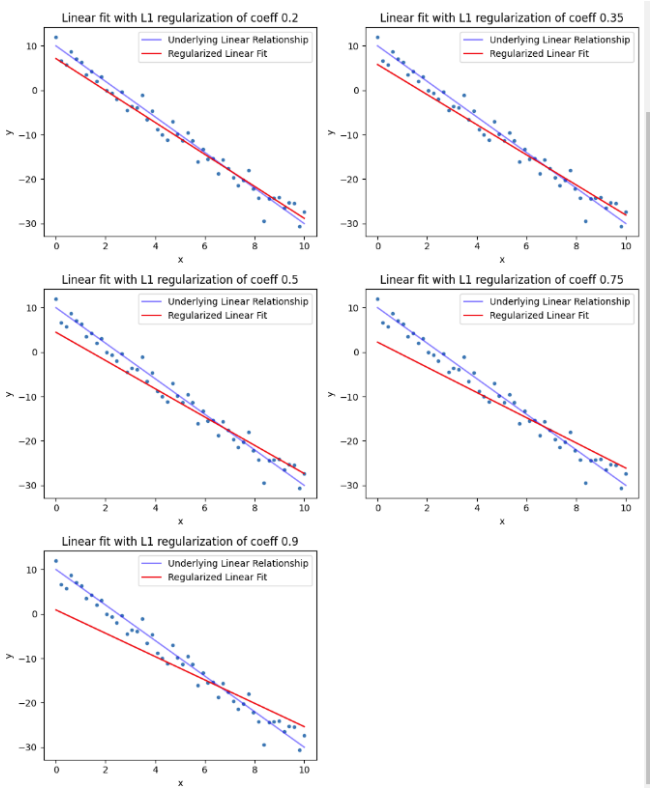

Figure 17: Linear Fit with Varying Lambdas (L1, lambda = 0.2 - 0.9) - Task 4

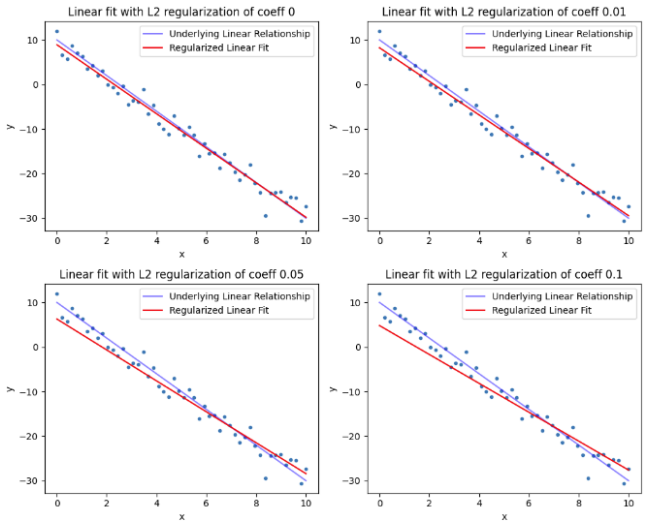

Figure 18: Linear Fit with Varying Lambdas (L2, lambda = 0 - 0.1) - Task 4

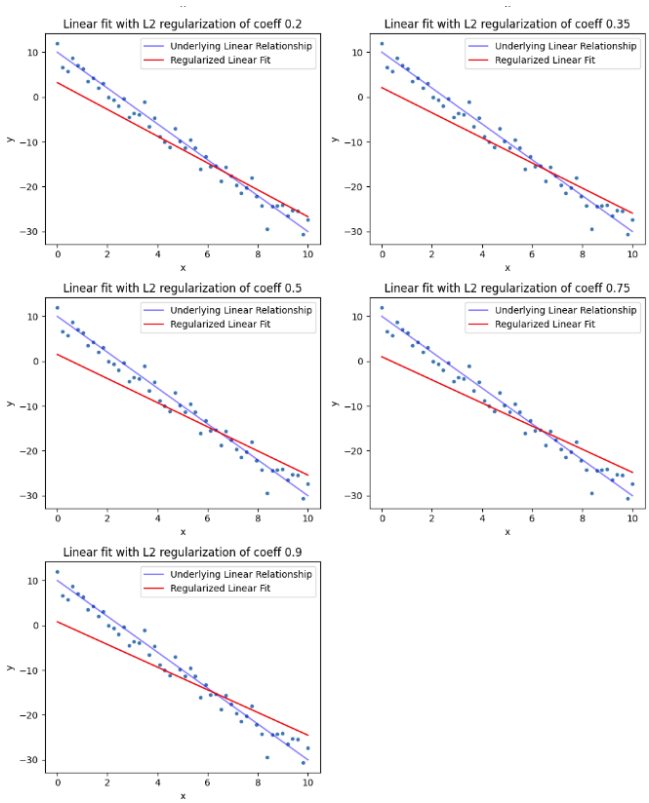

Figure 19: Linear Fit with Varying Lambdas (L2, lambda = 0.2 - 0.9) - Task 4

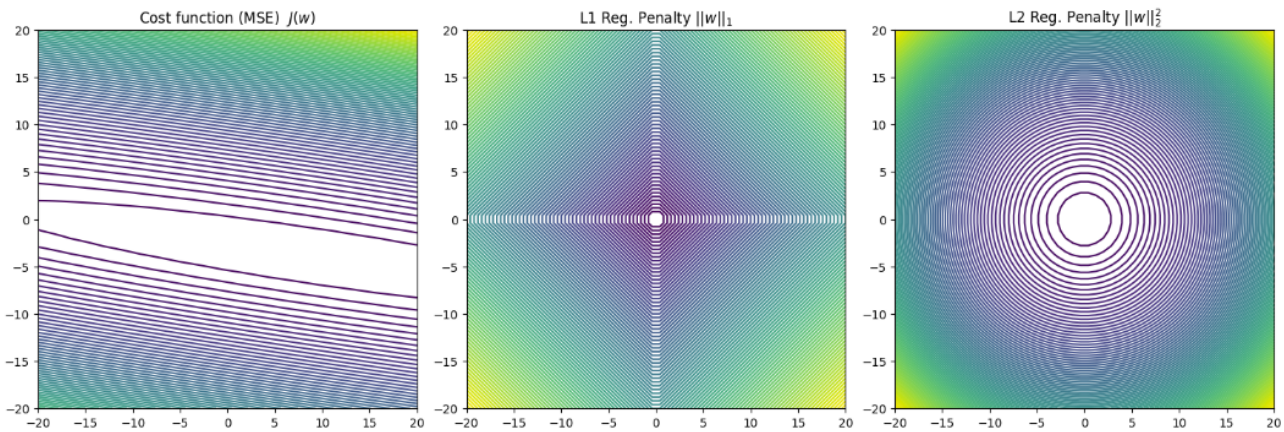

Figure 20: Visualizing Loss Function with L1 and L2 Regularization - Task 4

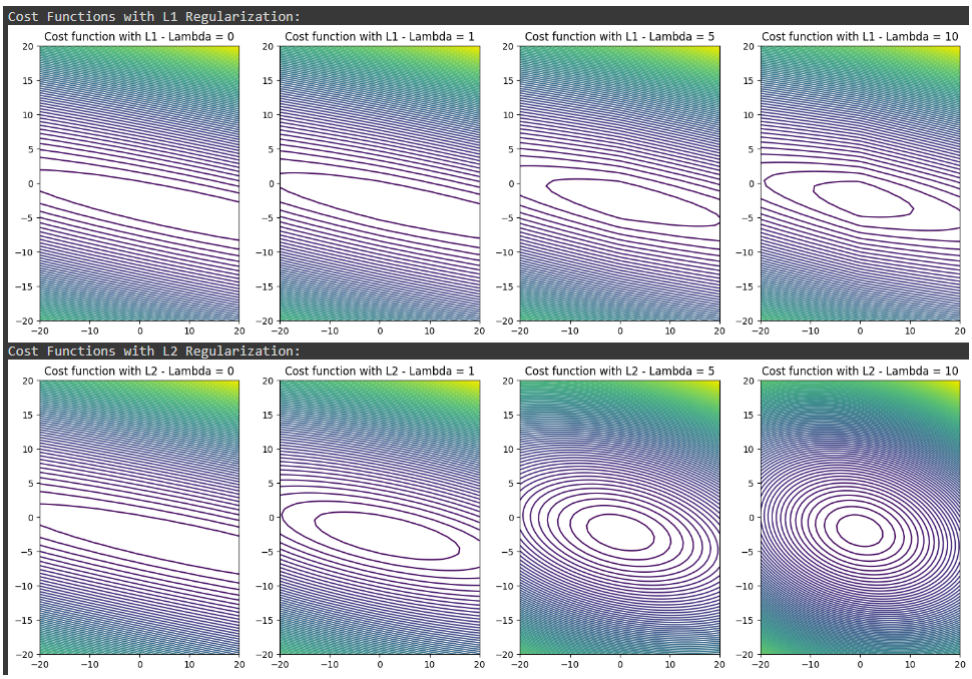

Figure 21: Effects of L1 and L2 Regularization on the Cost Function - Task 4

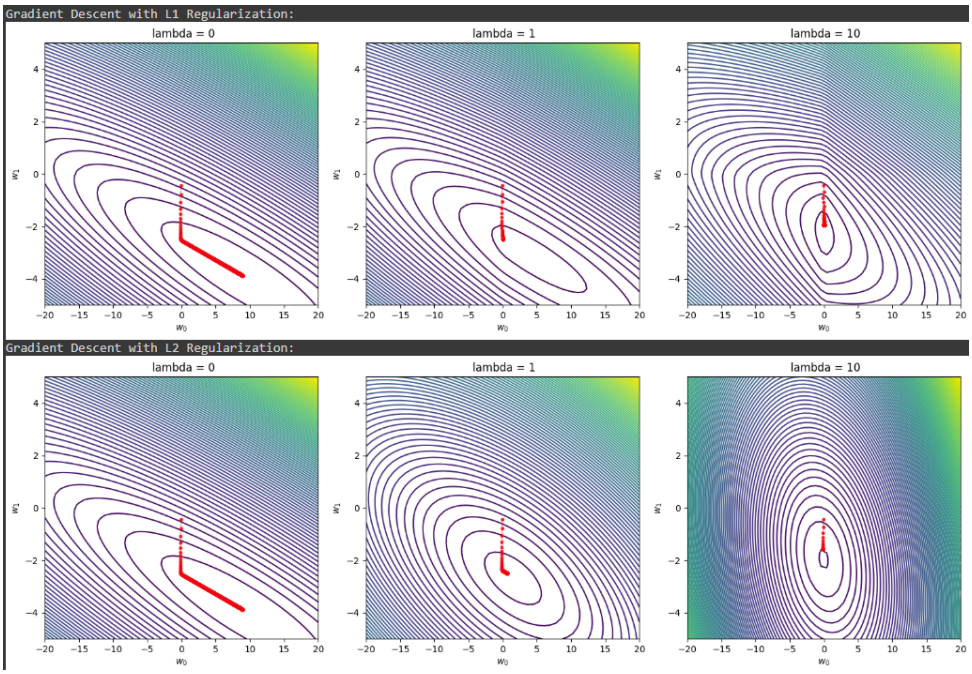

Figure 22: Effects of L1 and L2 regularization on Gradient Descent - Task 4

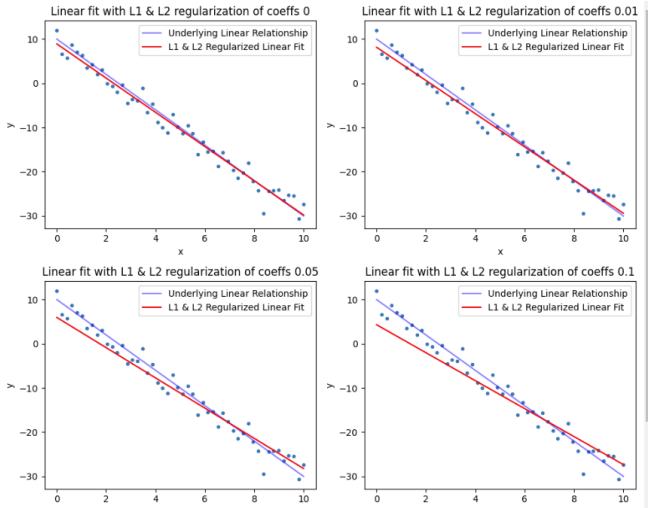

Figure 23: Linear Fit of model with Elastic Net Regularization from $\lambda = 0.0$ to $\lambda = 0.1$ - Task 5

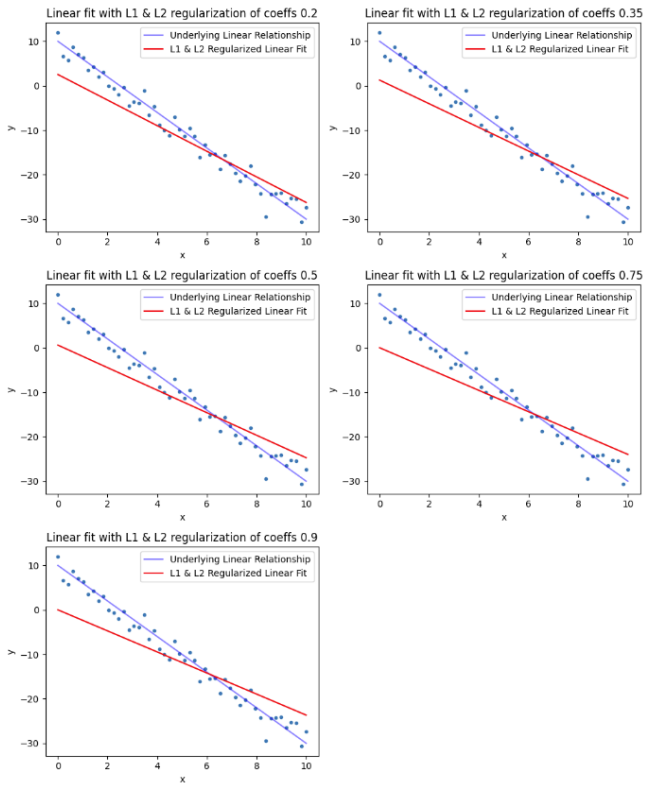

Figure 24: Linear Fit of model with Elastic Net Regularization from $\lambda = 0.2$ to $\lambda = 0.9$ - Task 5

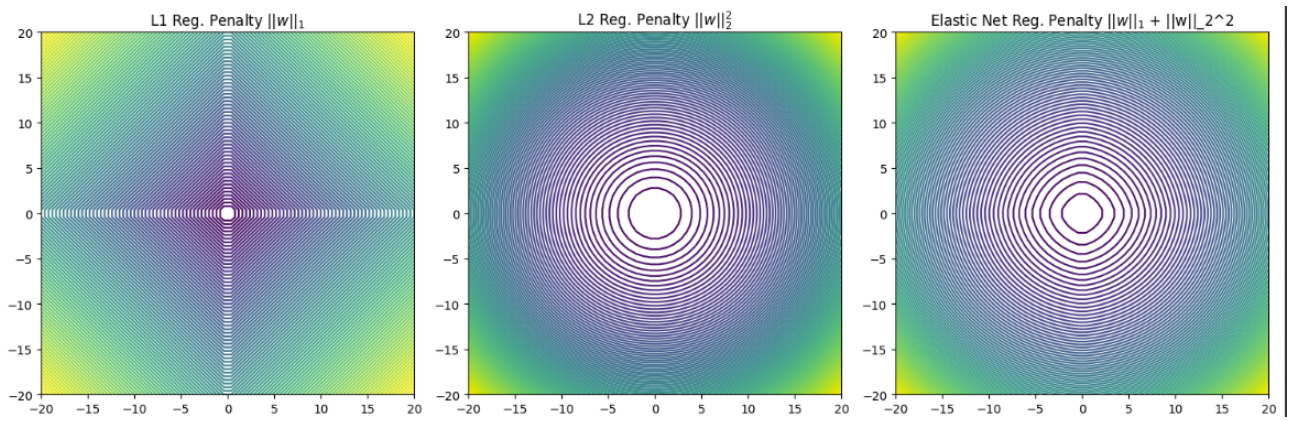

Figure 25: (Regularization Penalties) - Task 5

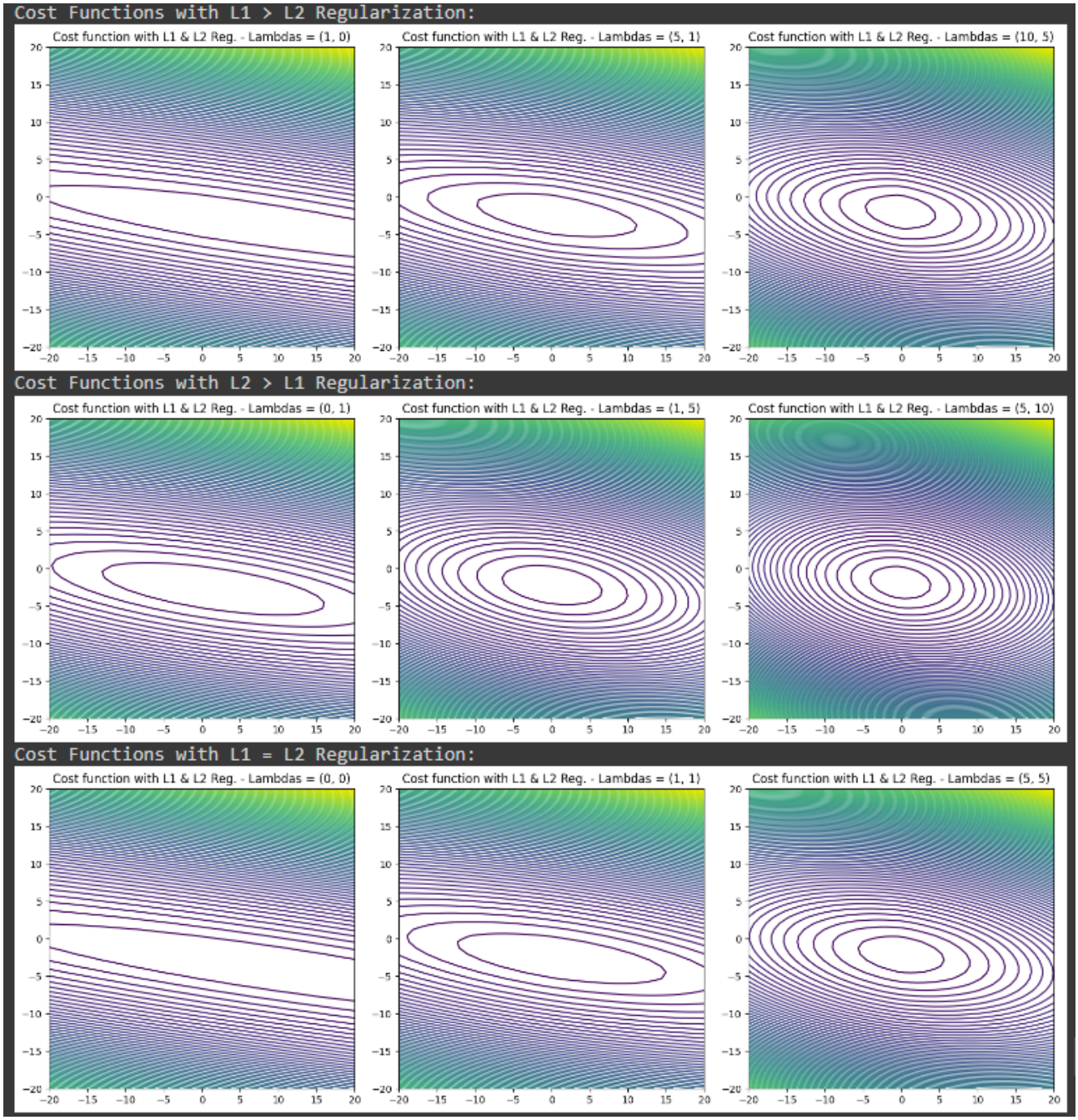

Figure 26: Effect on Loss Functions with various values of λ_1 and λ_2 - Task 5

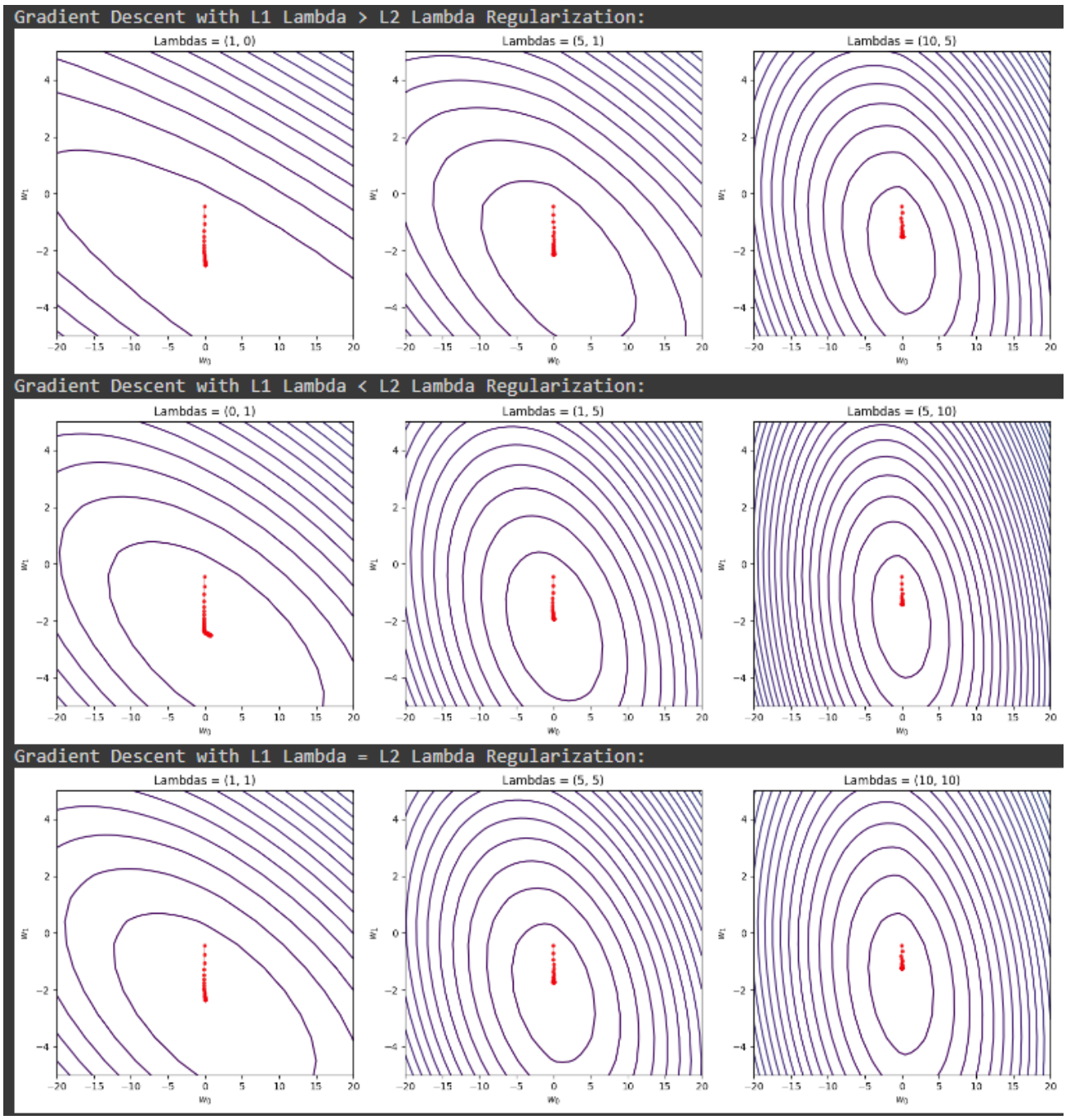

Figure 27: Gradient Descent with various values of λ_1 and λ_2 - Task 5