Assignment 2: Regularization and Model Evaluation

261250317 Léo Valette
261091403 Kayvan Dharsee
261111350 Kohei Okabayashi

Abstract

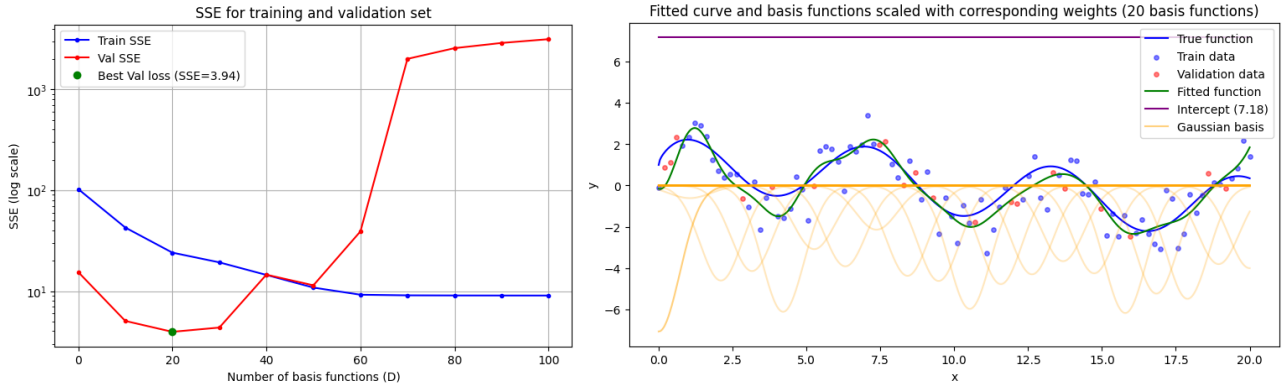
In this assignment, we investigated the effects of model complexity and regularization on linear regression using non-linear basis functions. By fitting models with varying numbers of Gaussian basis functions, we analyzed the bias-variance trade-off, identifying the optimal number of bases that minimizes the validation error. Additionally, we explored ridge and lasso regression to determine the best regularization strength, noting how L1 regularization encourages sparsity by driving weights to zero, while L2 penalizes large weights without inducing as much sparsity. We visualized these effects on contour plots, displaying how regularization can change the shape of the loss function and shorten the path of gradient descent. Through conducting hyper-parameter tuning and implementing optimization techniques such as momentum, we enhanced model expressiveness and performance. These efforts, combined with refined stopping conditions for gradient descent, led to more accurate predictions, effective regularization, and faster training convergence.

Task 1: Linear Regression with Non-Linear Basis Functions

In this task, we generated synthetic, non-linear data, and added various amounts of gaussian bases to a linear regression model to analyze how model complexity affects the fit of a model. We did this for 11 different amounts of bases, and plotted the results for analysis.

Initially, with very few basis functions (e.g. D = 0 or D = 10), the model is too simple, failing to capture the complex patterns present in the data. This results in a high Sum of Squared Errors (SSE) for both the training and validation sets, as the model cannot fit the true function accurately, leading to underfitting.

As we increase the number of basis functions (e.g. D = 20), the model becomes more capable of capturing the underlying trends, effectively reducing the SSE on the training data. This improvement is also reflected in the validation SSE, which decreases, indicating that the model is better at generalizing to new data. The lowest validation SSE, as observed in the SSE plot, is achieved with 20 basis functions, suggesting that this is the optimal model complexity. At this point, the model strikes a balance between flexibility and generalization, minimizing both underfitting and overfitting.



(a) Evolution of SSE for different value of D

(b) Best model (D=20)

Figure 1: SSE plot and best model fit for task 1

Beyond this optimal point, increasing the number of basis functions (e.g. D = 40 and higher) leads to overfitting. The model starts fitting to the noise in the training data rather than capturing the true function. This overfitting

is evident from the training SSE continuing to decrease, while the validation SSE starts to increase significantly, indicating that the model no longer generalizes well to new data. Another sign that we are overfitting is that the weights begin to get extremely large.

In summary, evaluating the model's performance on both training and validation sets, we can identify the point where the model transitions from underfitting to overfitting. The optimal model, with around 20 basis functions, provides the best balance between complexity and generalization, as indicated by the lowest validation SSE.

Task 2: Bias-Variance Tradeoff with Multiple Fits

In this task, we explored the bias-variance tradeoff by fitting models of varying complexity to the data and analyzing how they performed across multiple noisy datasets. The true function plot, overlaid with training and validation data points, illustrates the inherent randomness introduced by noise. This noise highlights the challenge of fitting a model that can accurately capture the true underlying trend.

The subplots for different values of D (number of basis functions) provided a detailed look at how model complexity affects bias and variance:

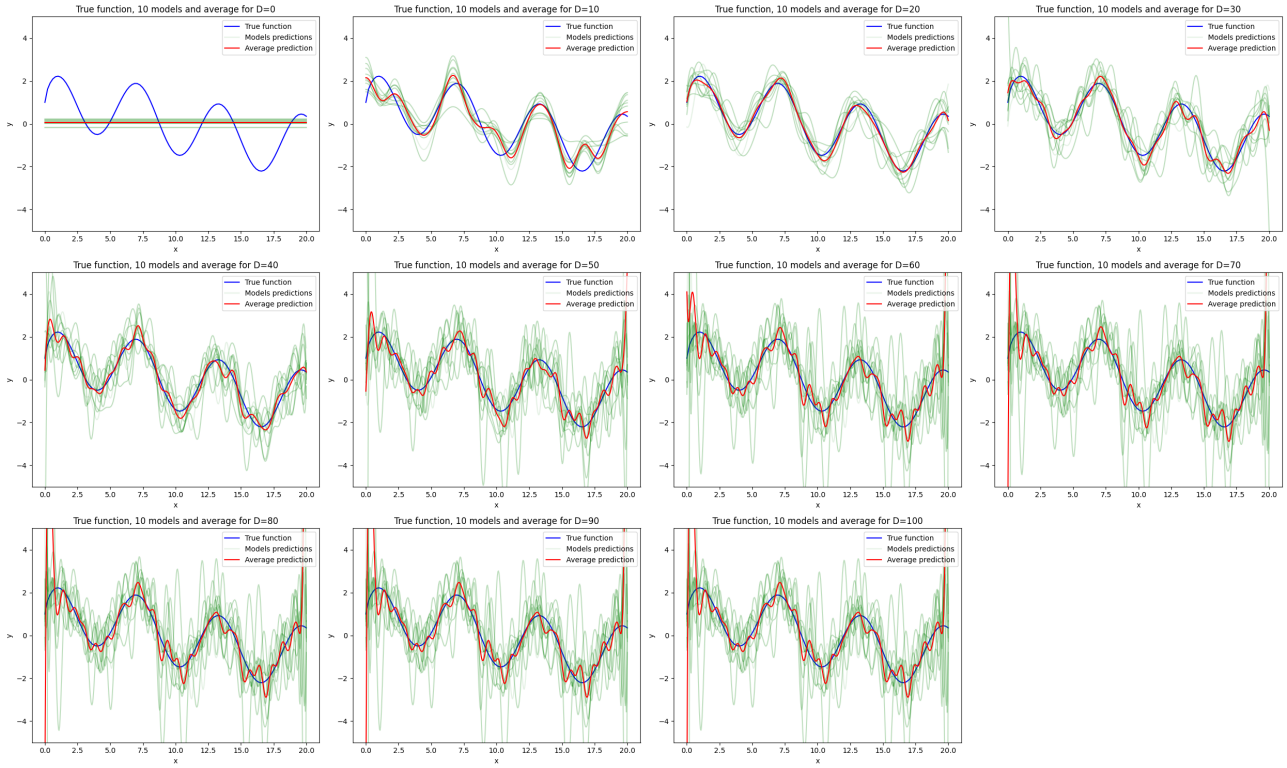


Figure 2: Model predictions and average performance across 10 models on increasing number of basis functions

Low complexity models (e.g. D = 0, 10) exhibited high bias, as they were unable to capture the true oscillatory nature of the data. The fitted lines were overly simplified and underfit the true function. While these models had low variance and were consistent across different noisy datasets, their inability to learn the true pattern resulted in poor SSE and MSE values for training and validations set.

Moderate complexity models (e.g. D = 20, 30) models struck a good balance between bias and variance. The fitted lines began to closely match the true function, and the average model aligned well with the actual trend. The variance was controlled, meaning the models did not fluctuate excessively across different datasets. These models were flexible enough to learn the pattern but stable enough to generalize well. The D = 20 model was especially good, providing the lowest SSE and MSE values for the the validation set, while having relatively low SSE and MSE values for the training set, indicating it is the optimal tradeoff and model.

High complexity models (e.g. D = 40 to 100) were highly flexible, which resulted in high variance. The individual models fluctuated significantly across different noisy datasets, a sign of overfitting. While these models exhibited low bias and were able to fit the training data closely, they also captured the noise within the data, which made

them less reliable when predicting new, unseen data. Although the red line of the plots seems to indicate that the bias begins to raise as the number of bases increases after a certain point, this is not indicative of what actually occurs. The bias does indeed lower as model complexity is increased, and we address why the plot seems to show otherwise in the originality section of the report.

The SSE and MSE values for training and test sets provided us a quantitative way of comparing the performance of these models, which allowed us to determine an optimal tradeoff/model.

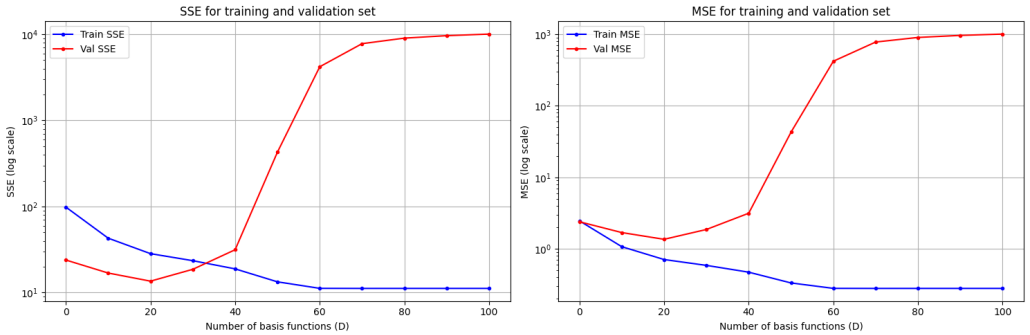


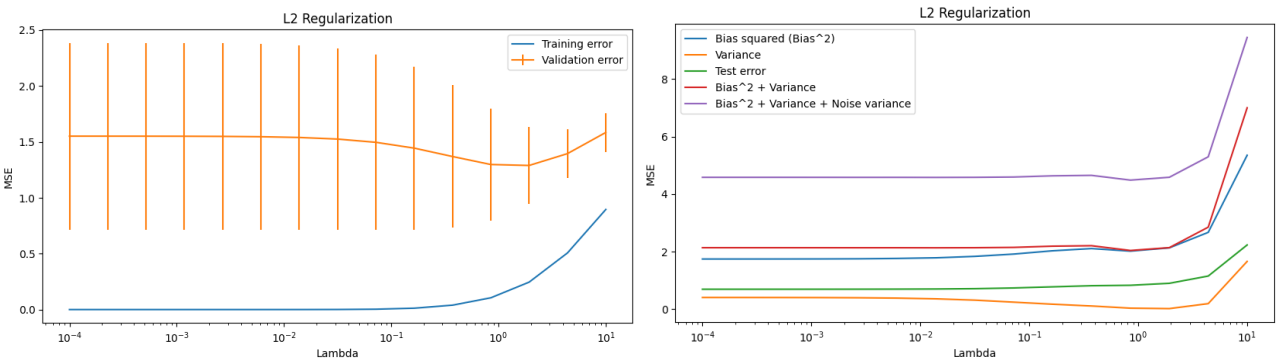
Figure 3: SSE vs MSE of training and validation sets on increasing number of basis

Initially, both training and test errors decreased as the number of basis functions D increased, indicating that more complex models were effective in reducing bias. However, beyond a certain level of complexity, the test error began to rise, while the training error continued to decrease. This was the beginning of overfitting, where the model not only learns to fit the training data perfectly but also captures the noise, resulting in poor generalization to new data.

In summary, the best performing model lies somewhere in the middle, specifically where the validation error is minimized, striking a balance that allows for generalization to new data.

Task 3: Regularization with Cross-Validation

Based on the validation errors, the optimal lambda for ridge regression is approximately 1.5, while for lasso regression, it is around 0.1. We determined these values by limiting the maximum possible lambda in our range of options, as higher values led to significantly higher Mean Squared Errors (MSE), making it difficult to identify the minimum effectively in the plots. When comparing this to the optimal lambdas based on the generalization error, we find that the optimal value remains the same for ridge regression but changes to 0.85 for lasso regression. The reason for this difference is that our test error serves as an estimate of the generalization error, since in most cases, we do not have access to the original distribution needed to compute the true generalization error. However, as we do have the generalization error in this instance, it provides a more accurate indicator of the optimal lambda values for both forms of regularization.



(a) Training and validation MSE

(b) Generalization error and test error

Figure 4: L2 regularization error plots

For both ridge and lasso regression, larger values of lambda, such as 10, result in sharp increases in both bias and variance, leading to a significant rise in error. This occurs because strong regularization forces the weights to

be extremely small, preventing the model from fitting the data well Murel & Kavlakoglu (2024). Before reaching this point of sharp increase, we observe that as lambda increases, variance slightly decreases while bias slightly increases. It is in this range that we find the minimum of the generalization error, indicating the point at which the model performs best.

Task 4: Effect of L1 and L2 Regularization on Loss

We can observe how L1 regularization encourages sparsity by examining the effect of increasing lambda on the loss contours. With a higher lambda, the loss contours become centered closer to zero on both weight axes, and their area shrinks. This makes it so the area of which the loss function is minimized becomes smaller, causing the optimal weights to be close to zero. Additionally, L1 regularization alters the contours shape, causing them to obtain a diamond-like shape rather than remain circular. This change makes it more likely that the optimal weights will fall on the zero axes. As a result, the model's optimal weights are often very close to zero or exactly zero, leading to a sparse model.

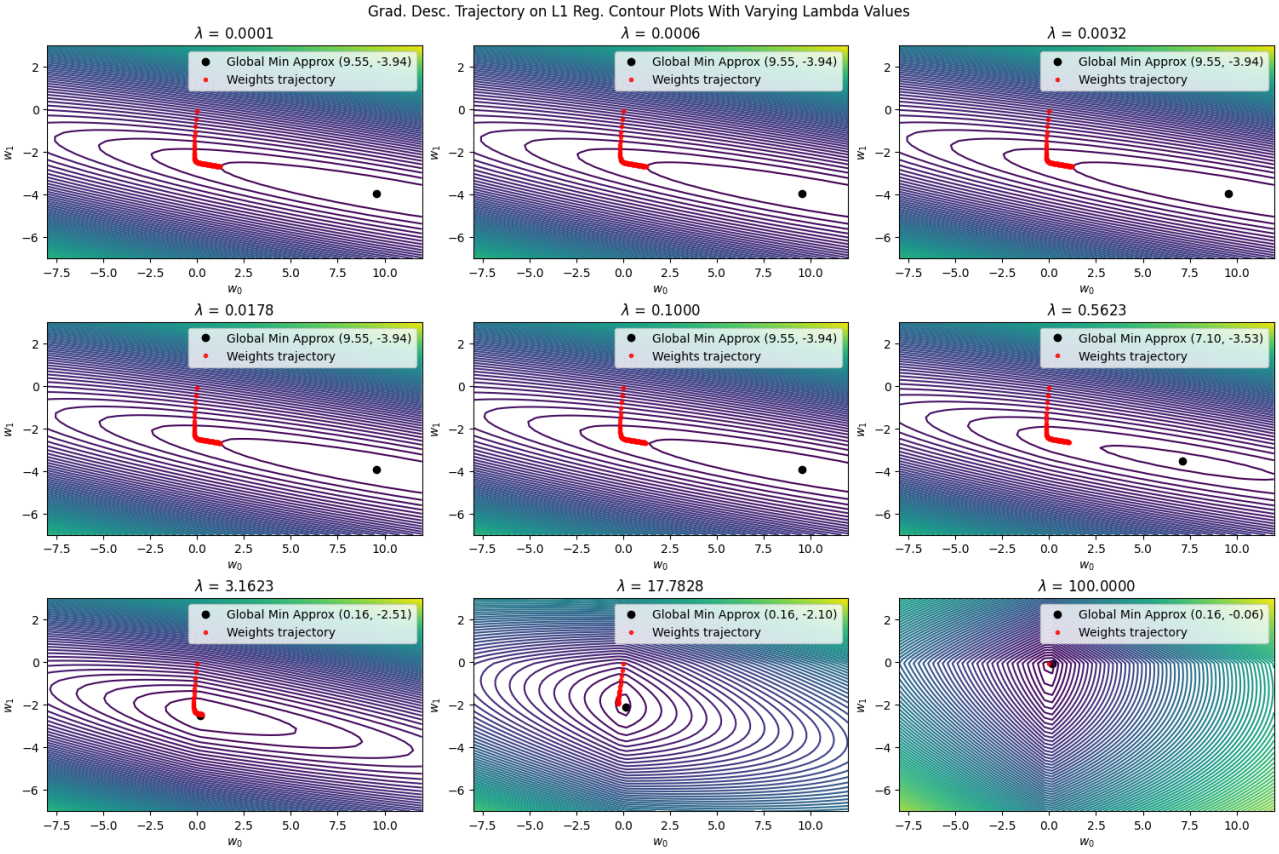


Figure 5: Gradient descent trajectory on L1 regularization contour plots with varying lambda values

L2 regularization also penalizes large weights by adjusting the loss contours to be more centered around zero and reducing their area. However, unlike L1, L2 regularization retains the circular shape of the contours. This circular form makes it less likely to produce weights that are exactly zero as there is more area for the optimal weights to fall away from the zero axes. It is this difference in contour shape which causes L2 regularization to not promote sparsity as strongly as L1.

As previously mentioned, increasing the value of lambda centers the loss contours closer to zero, shrinks their area, and can alter their shape depending on the type of regularization. By centering the contours around zero, the optimal weights are pushed closer to zero. This reduces the number of steps needed to reach the optimal weights, as training typically begins with weights set to zero. Consequently, this leads to shorter optimization paths, allowing for faster training.

Originality/Creativity

To go beyond the basic requirements, we conducted several experiments to enhance the expressiveness and effectiveness of the linear regression model. We conducted manual hyper-parameter tuning and implemented optimization techniques, such as adding momentum to gradient descent. This allowed for our model to be more expressive, which helped highlight the effects of regularization.

Additionally, we included extra data points strictly for plotting purposes, allowing us to generate smoother graphs that more accurately displayed the results. We also experimented with penalizing the intercept to determine whether it would improve the regularization effect, ultimately deciding on not doing so. We also computed the closed form solution using various different functions such as inv, pinv, and lstsq from the numpy library. This allowed us to visualize how their utilization affected our model's performance on the train set using the SSE loss, and select which function that we were allowed to use provided us with the best performance, which was the pinv function.

In order to calculate the bias, we need the average model over infinite datasets, which is not feasible. When we have low variance in our model, the bias calculated from a few datasets gives an accurate approximation of the true bias. However, this does not hold as variance increases. To accurately visualize our bias, we produced plots using a higher number (N=500) of datasets to allow for a more accurate line to represent the true bias of the model. We ensured that we used different train/validation splits across every dataset to allow for true random sampling to provide evenly distributed data to both sets. Finally, we tested various stopping conditions for gradient descent and found that using the difference between the previous iterations loss function and the current iterations loss function, rather than the norm of the gradient, led to more effective model training.

Discussion and Conclusion

In this assignment, we solidified and expanded our understanding of regularization, generalization, and model expressivity. We experimented with adding gaussian bases and momentum to linear regression to allow our model to fit non-linear data and improve its expressivity. We then used plots to select the optimal number of gaussian bases in order to avoid overfitting and underfitting, and manual hyper-parameter tuning to select our momentum value. We also experimented with both ridge and lasso regression across various values of lambda, testing on both the gradient descent method and the analytical method where applicable. As we were working with synthetic data, we were able to compute and view the bias, variance, and thus the generalization error of our model. This combined with viewing the test error using k-fold cross-validation allowed us to tune hyper-parameters effectively by finding the optimal settings for these hyper-parameters, such as the regularization lambda value.

There are still numerous other avenues for further exploration in this assignment. Although we experimented with L1 and L2 regularization, there are other p-norms that could be explored. For instance, optimizing L0 regularization would be an interesting direction, though it presents more challenges due to the non-convex nature of this specific p-norm. We could also have experimented with different cross-validation techniques, such as leave-one-out cross-validation, to compare which method provides a validation error that better estimates the generalization error.

Statement of Contributions

We all wrote our own code and reports, met up together to discuss different experiments, results, and methods to display our results. We chose what worked best from each of our reports to create a final notebook and report. Kayvan's strategies and insights along with Léo's knowledge of the models and creative ideas for making better plots were the highlights of the assignment, and Kohei worked to summarize it all together at the end to finalize the report.

References

Murel, J., & Kavlakoglu, E. (2024, September). *What is regularization?* IBM. Retrieved from https://www.ibm.com/topics/regularization#:~:text=Regularization%2C%20however%2C%20can%20potentially%20lead,i.e.%20models%20with%20few%20parameters (Accessed: 2024-10-23)

Appendix

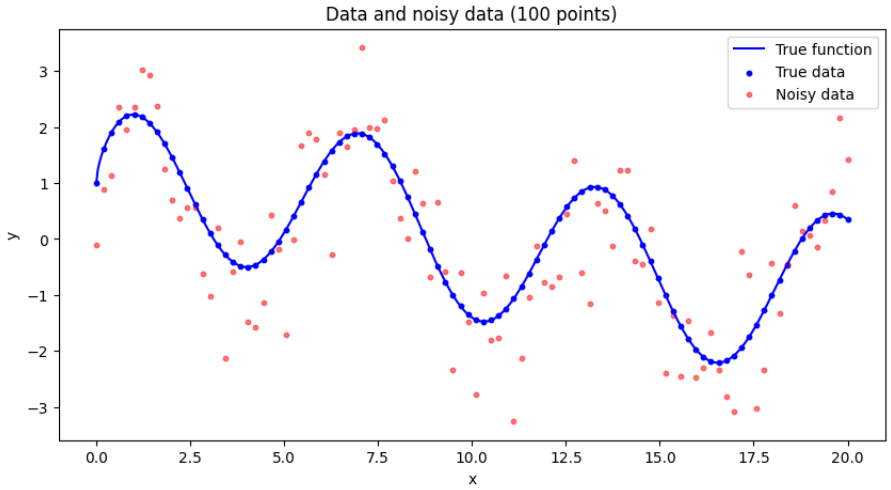


Figure 6: Training data for Task 1

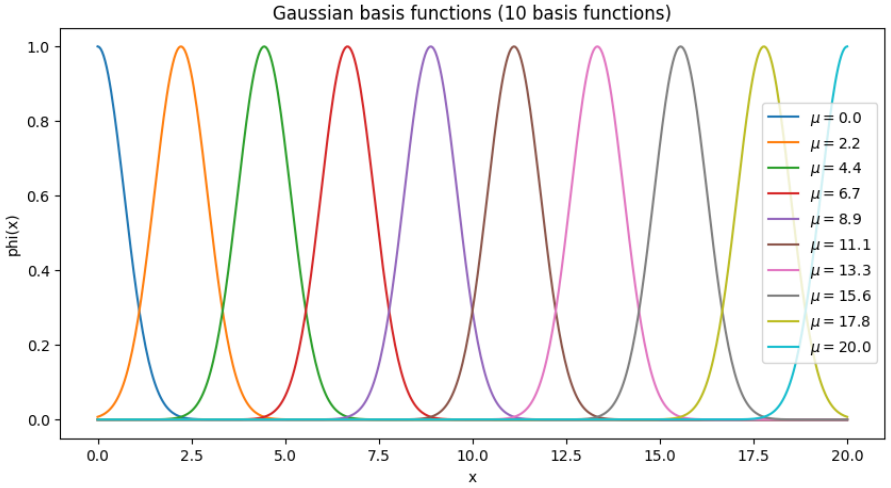


Figure 7: Unweighted gaussian basis functions

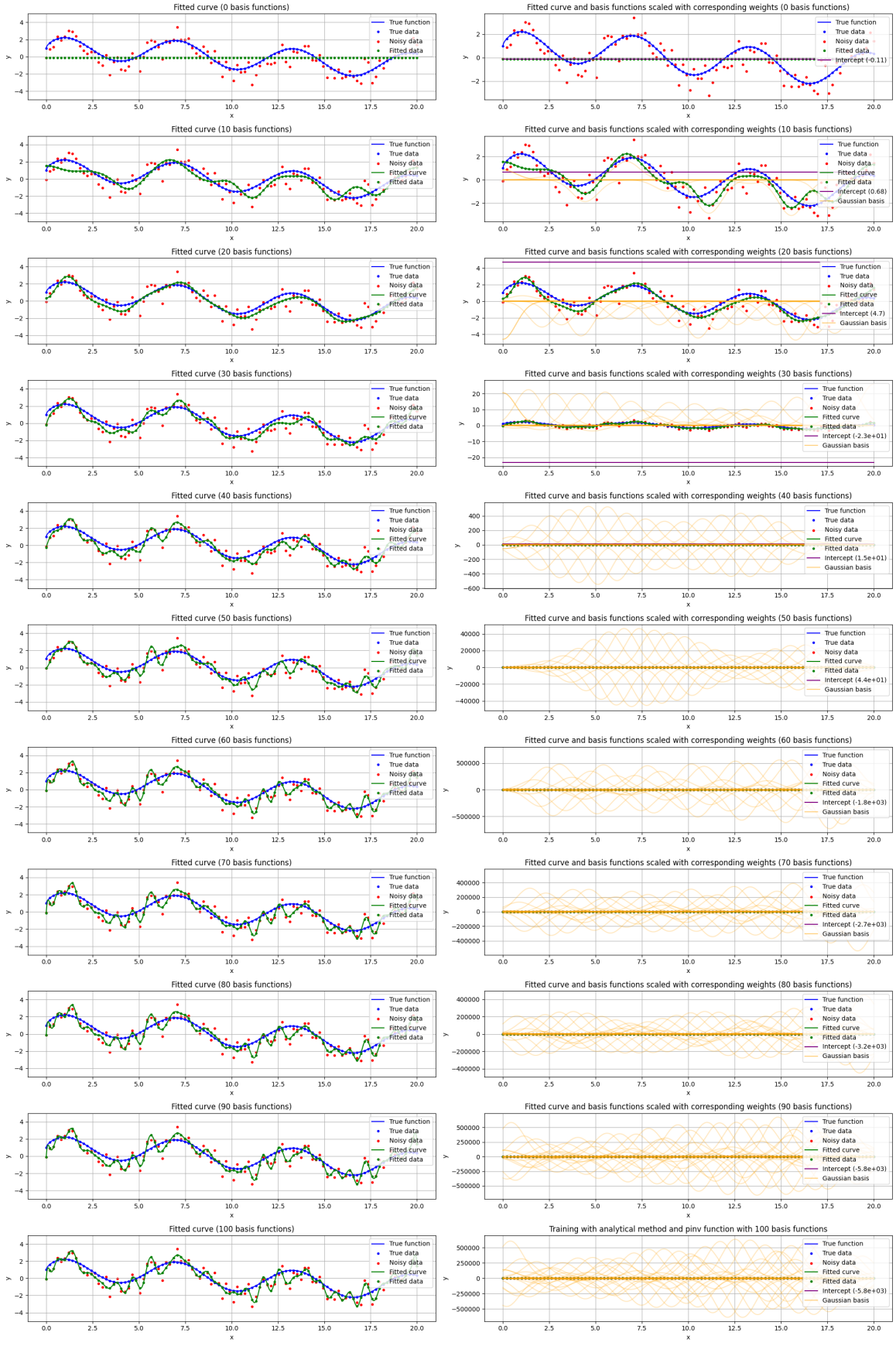
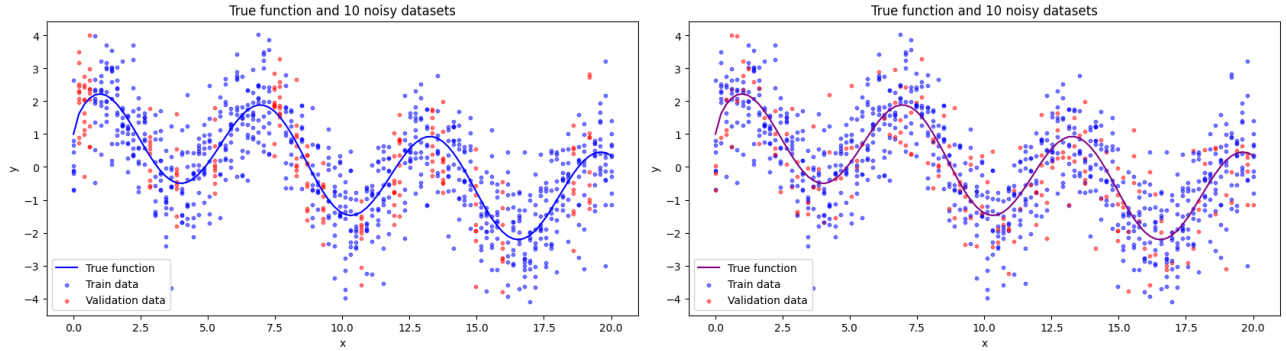


Figure 8: Effect of increasing basis functions on curve fitting

Figure 9: Comparison of different methods for analytical linear regression



(a) Version 1

(b) Version 2

Figure 10: Comparison of different methods of interspersing validation and training data

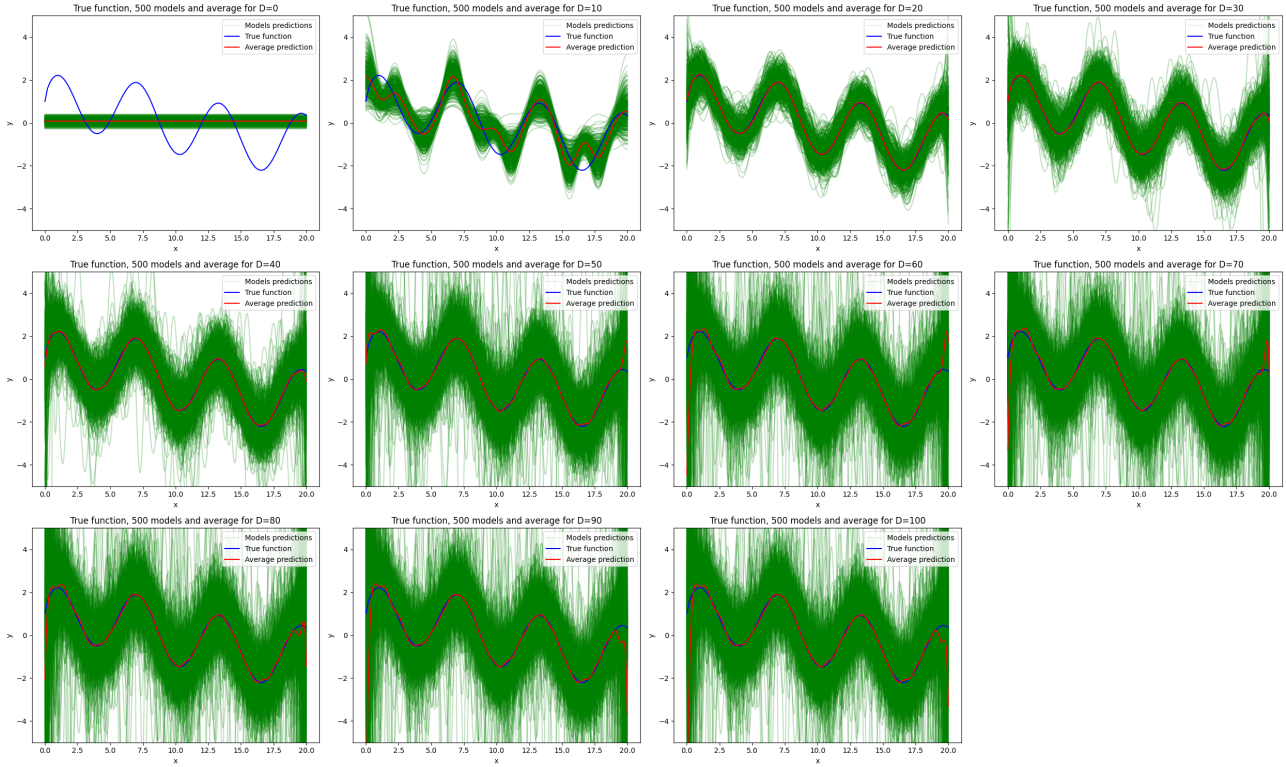
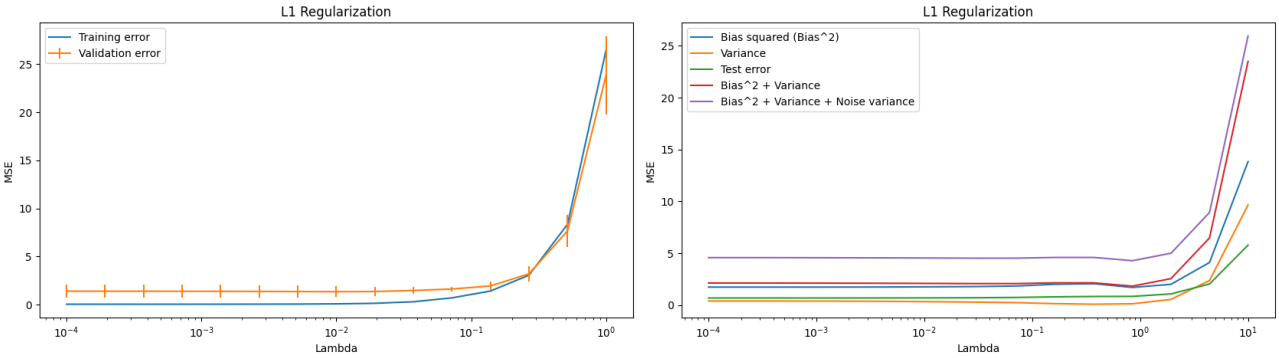


Figure 11: Variation in model predictions across 500 trials for different amounts of gaussian bases



(a) Training and validation MSE

(b) Generalization error and test error

Figure 12: L1 regularization error plots

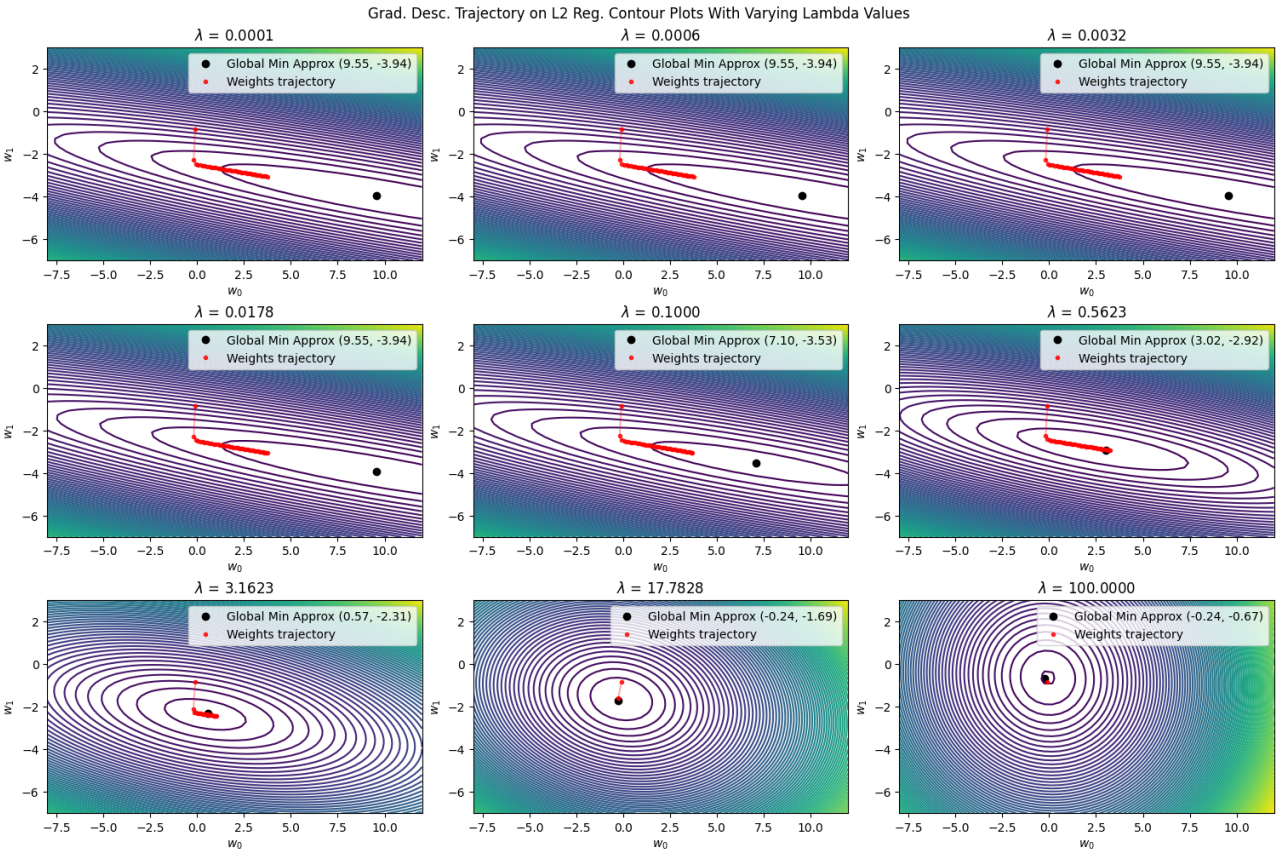


Figure 13: Gradient descent trajectory on L2 regularization contour plots with varying lambda values