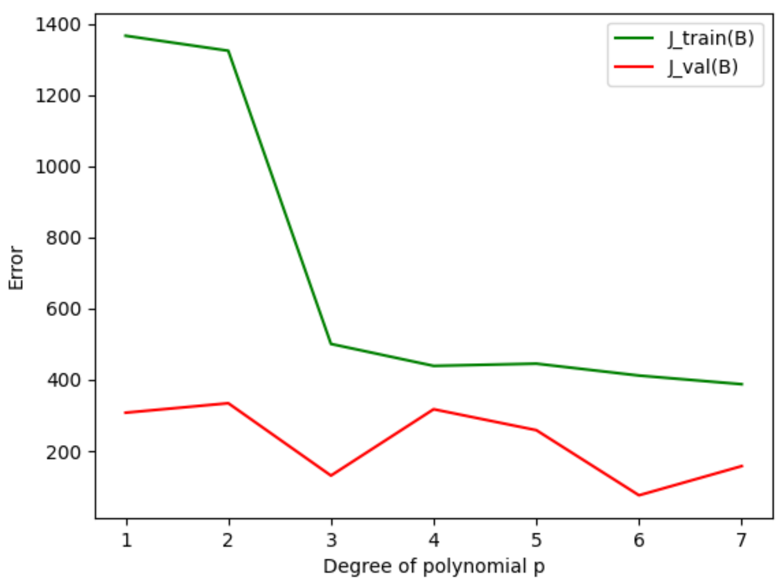
I was confused what datasets we were using for this assignment. I generated code that created 7 polynomial datasets from 1, 7. So the first was just some linear data while the last was data representing a polynomial of degree 7. Then I trained each model based on teh respective polynomial. I then plotted the training data (blue), resulting model (red) and training, validation, and test error as shown in figure 1. It makes sense that each polynomial model performs well since they were individually trained on their respective p degree data set.

Then, I realized that maybe this isn't the point of the assignment. Intuitively, what would make more sense as an assignment is to train the seven different polynomial models on one data set, say, a dataset representing a degree 5 polynomial. I am leaning this way as well considering the example regards a degree 5 polynomial and dataset.

This would make so much more sense for an assignment because it will result in directly seeing how underfitting and overfitting results in the error graphs. For example, if we try and train a degree 1 polynomial and on the degree 5 poly data set, it will result in underfitting. On the other hand, if we train a degree 7 polynomial on a degree 5 dataset, it **may** result in overfitting. Figure 2 displays that data.

Below is a plot of figure 1: Model selection procedure from the assignment. On the X axis we have degree of polynomial p and the Y axis is the error. This is for the degree 5 polynomial dataset described above. It makes sense that we have a very low training error for degree 1 and 2 polynomials against a degree 5 dataset. I used the cost at the last epoch as the error term for each polynomial degree.



Unfortunately this plot looks nothing like the actual plot in Figure 1 despite the fact that it theoretically should look the same considering what I described above. Its surprising to me that the validation cost is always smaller than the training cost despite using different polynomials. Nonetheless, to answer the question, **I would choose a degree 6 polynomial because it results in the lowest validation cost and the second lowest training error.**

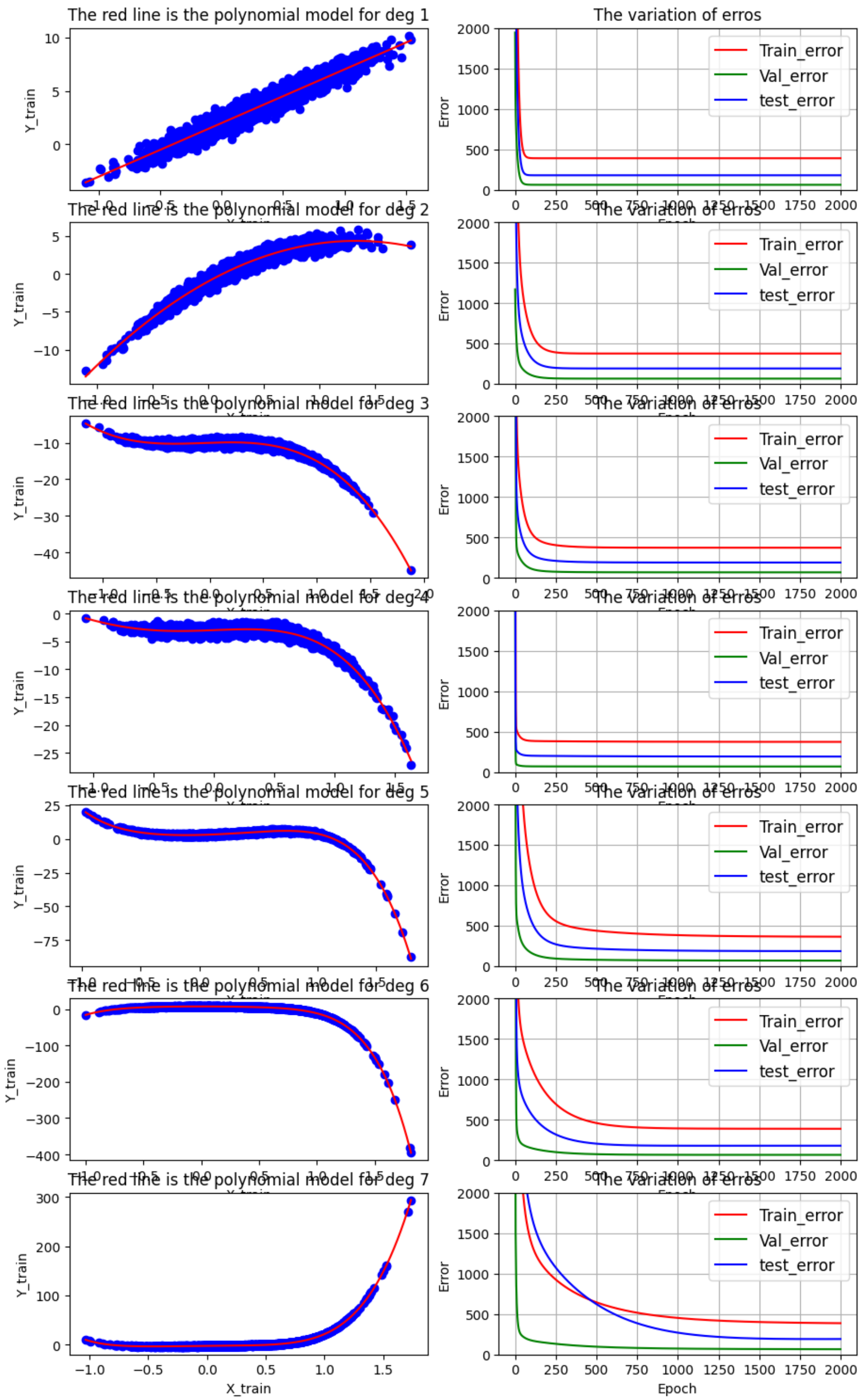


Figure 1: Data for training a degree p polynomial against a degree p polynomial data set

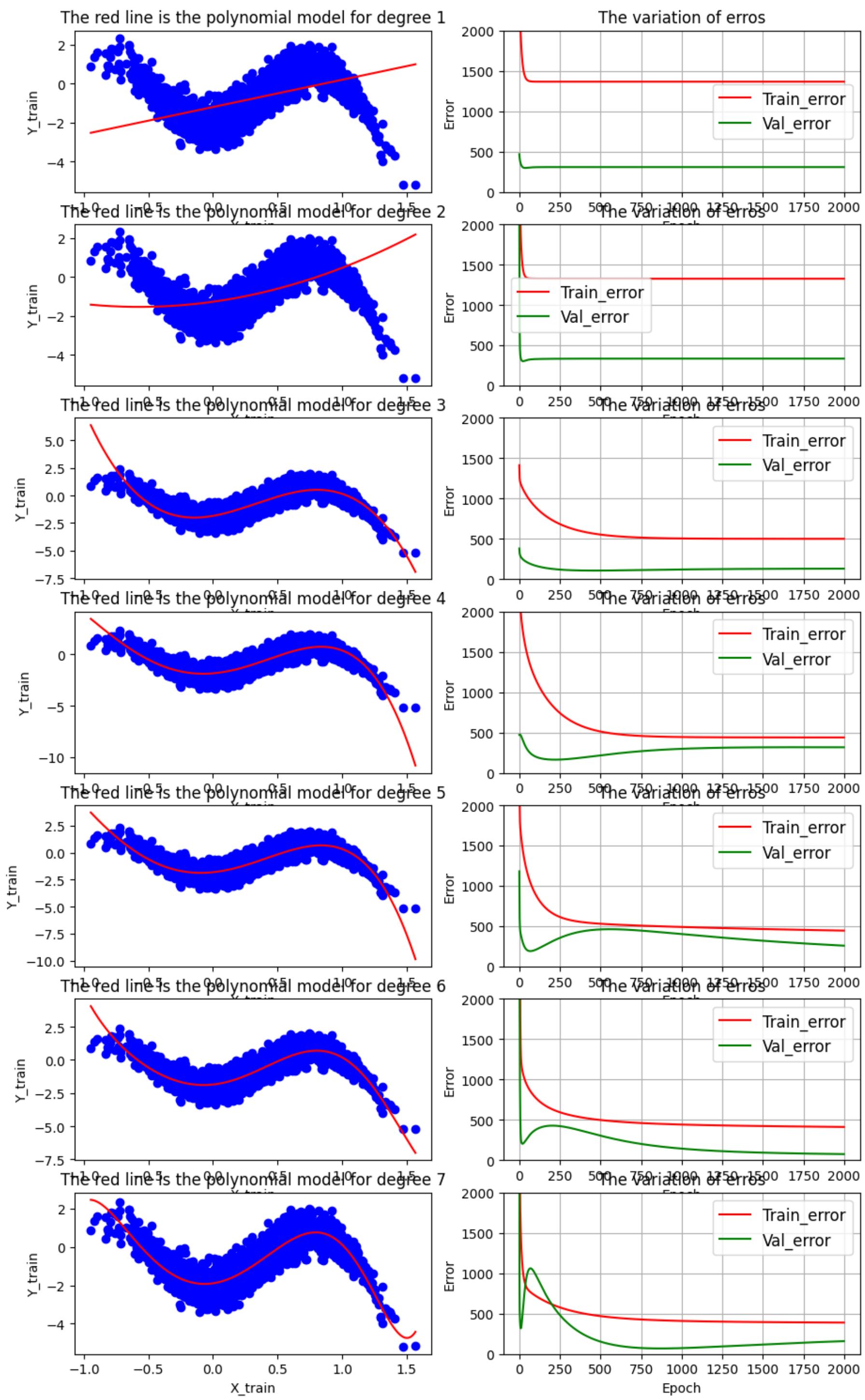


Figure 2: Data for training a degree p polynomial against a degree 5 polynomial data set

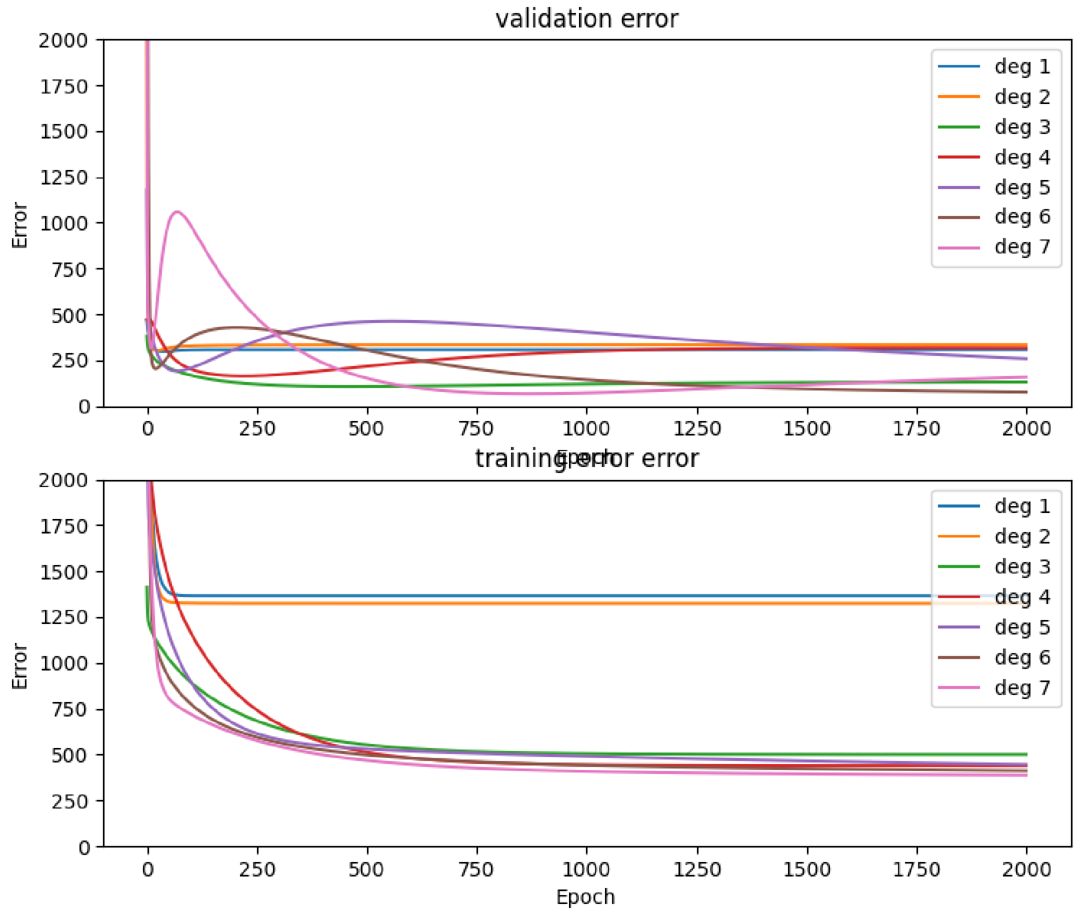


Figure 3: Data for training costs and validation costs at each iteration for all 7 polynomials degrees trained against p =5 dataset.