

Part 2: Make predictions by using your implementation

Here are the results for various combinations of alpha (learning rate) and “n_epoch” (number of epochs):

```
D:\Anaconda\python.exe "D:\WPI Assignments\Machine Learning CS539-F24\Homeworks\Homework3\Assignment3\application.py"
Alpha: 0.01, Epochs: 20, Training Loss: 4173.283218, Testing Loss: 4433.483239
Alpha: 0.01, Epochs: 30, Training Loss: 3516.274259, Testing Loss: 3762.132353
Alpha: 0.01, Epochs: 40, Training Loss: 2966.748129, Testing Loss: 3195.180019
Alpha: 0.01, Epochs: 50, Training Loss: 2506.243709, Testing Loss: 2715.784150
Alpha: 0.01, Epochs: 100, Training Loss: 1095.245420, Testing Loss: 1216.403728
Alpha: 0.1, Epochs: 20, Training Loss: 196.013791, Testing Loss: 225.415379
Alpha: 0.1, Epochs: 30, Training Loss: 39.494947, Testing Loss: 46.302882
Alpha: 0.1, Epochs: 40, Training Loss: 8.317329, Testing Loss: 9.925180
Alpha: 0.1, Epochs: 50, Training Loss: 1.824303, Testing Loss: 2.218352
Alpha: 0.1, Epochs: 100, Training Loss: 0.001420, Testing Loss: 0.001885
Alpha: 0.3, Epochs: 20, Training Loss: 0.200160, Testing Loss: 0.254013
Alpha: 0.3, Epochs: 30, Training Loss: 0.002122, Testing Loss: 0.002842
Alpha: 0.3, Epochs: 40, Training Loss: 0.000025, Testing Loss: 0.000034
Alpha: 0.3, Epochs: 50, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 0.3, Epochs: 100, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 0.5, Epochs: 20, Training Loss: 0.000127, Testing Loss: 0.000176
Alpha: 0.5, Epochs: 30, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 0.5, Epochs: 40, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 0.5, Epochs: 50, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 0.5, Epochs: 100, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 1.0, Epochs: 20, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 1.0, Epochs: 30, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 1.0, Epochs: 40, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 1.0, Epochs: 50, Training Loss: 0.000000, Testing Loss: 0.000000
Alpha: 1.0, Epochs: 100, Training Loss: 0.000000, Testing Loss: 0.000000

Best Parameters:
Best Alpha: 1.0
Best Epochs: 50
Best Training Loss: 0.000000
Best Testing Loss: 0.000000
```

Relationship Between Alpha and Number of Epochs:

- **Small Alpha (e.g., 0.01):**
When the learning rate is small, the model learns more cautiously, taking smaller steps during training. This means it needs to go through the training data (epochs) more times to improve. As a result, it converges very slowly, and even after many rounds, the testing loss still remains relatively high.
- **Medium Alpha (e.g., 0.1- 0.3):**
A moderate learning rate strikes a good balance. The model learns efficiently without overshooting, and as the number of epochs increases, it steadily reduces the errors in both training and testing. With the right number of epochs, the losses drop to very low levels, giving good results without needing too many epochs.
- **Large Alpha (e.g., 0.5- 1.0):**
A high learning rate means the model learns very quickly, reducing the errors significantly in just a few epochs. The training and testing losses reach close to zero in a small number of epochs (like 20 or 50). However, if the learning rate is too high, it could potentially make the model unstable or miss the optimal point, but that didn't happen in this case.

Summary:

Increasing alpha speeds up convergence, reducing the number of epochs required for the model to minimize loss effectively. However, extremely large values could potentially cause overshooting if not controlled well.