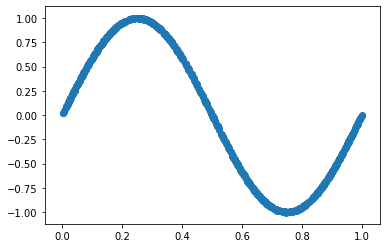
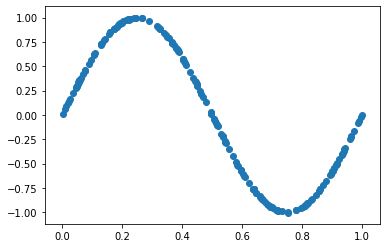
1. **Understanding the data and simple curve fitting**

a. Plot a feature vs label graph for both the training data and the test data.

Training Data



Test Data



b. Write a code to fit a curve that minimizes squared error cost function using gradient descent (with learning rate 0.05), as discussed in class, on the training set while the model takes the following form y = W Φ (x) , where , and T n W ∈ R n+1 Φn(x) = [1, x, x , x ... , x ] . Squared error is defined as . 2 3 n J(W) = 1 2m ∑ m i=1 (W Φ (x) T n − y) 2 In your experiment, vary n from 1 to 9. In other words, fit 9 different curves (polynomials of degree 1, 2, …, 9) to the training data, and hence estimate the parameters. Use the estimated W to predict labels on test data and measure squared error on the test set, name it as test error.

Following are for 500 iterations -

Train error

N=1 0.22400660601527023,

N=2 0.22633825456575274,

N=3 0.21499808821999897,

N=4 0.18713625541958398,

N=5 0.1587282713573692,

N=6 0.13645314773851017,

N=7 0.12095506504306766,

N=8 0.11103226638357266,

N=9 0.10522181122311418

Test error

N=1 0.19110484800361963,

N=2 0.1937088338226491,

N=3 0.18410356780377202,

N=4 0.16098846626590885,

N=5 0.13812743341230138,

N=6 0.12031377765559906,

N=7 0.10787256905059045,

N=8 0.0999968798138641,

N=9 0.09565787858340992