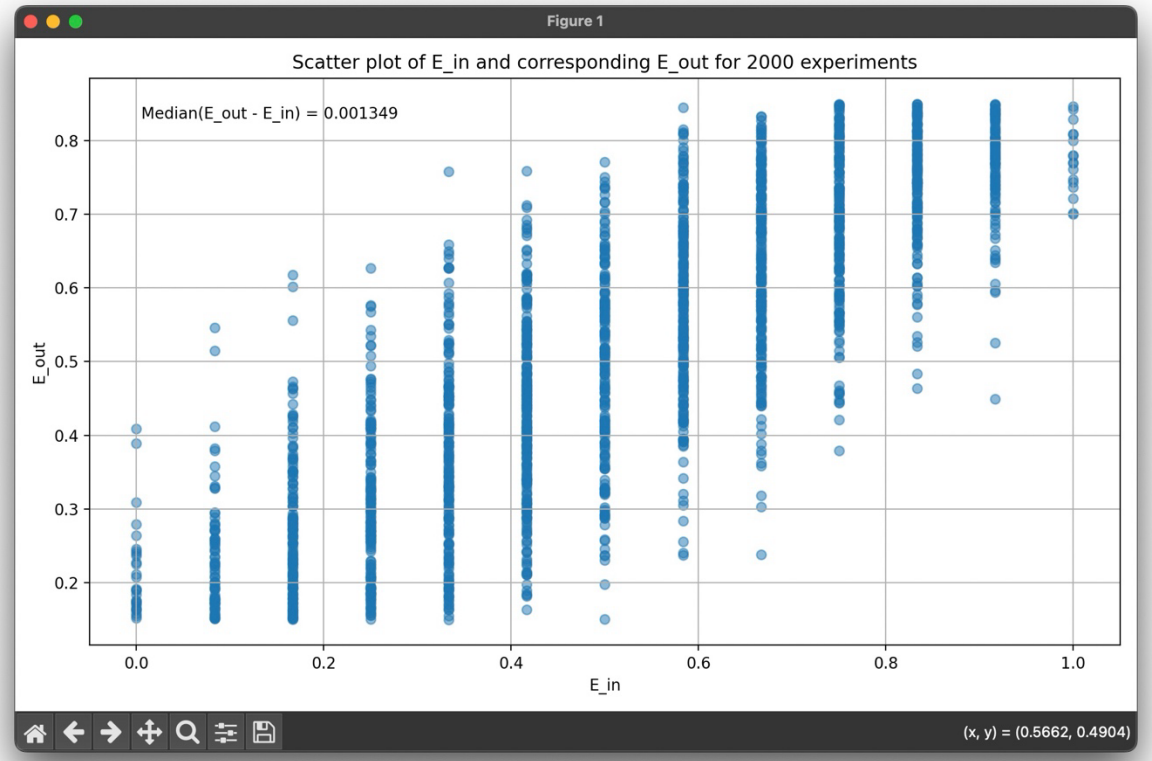
ML homework 2: question 12



Findings:

The median of $E_{out}(g_{RND}) - E_{in}(g_{RND})$ is about 0.001349, which is smaller than the resulting median of the previous question (0.088744).

I think the reason is because if we choose the optimal s and theta by finding the minimum in sample error, this will cause overfit to the data points, and while we only have 12 points in the dataset, this problem will become more severe, since it is hard to get a generalized result from merely 12 points, and if there's an extreme point (a point affected by noise), it will affect the hypotheses drastically. The bigger median of difference is telling that even though our model is performing well on the training data, it performs poorly on unseen data.

On the other hand, if we use randomly chosen s and theta, we introduce more randomness, probably having less probability to overfit, therefore making the result more generalized, so that the difference between in sample and out of sample error is smaller on average.

Snapshot of code:

Most of the code are the same as question 11, so I will provide the snapshot of the content that differs from the previous question. The first few 35 lines are the basic setup, which is the same as the previous question.

The modification is mostly presented in line 39, 40 in this snapshot, which shows that we randomly choose the value of s and theta, instead of calculating the minimum in sample error then decide the optimal s and theta:

```
36      data_points_list = list(zip(x_arr, y_with_noise_arr))
37      sorted_data_points_list = sorted(data_points_list, key=lambda point: point[0])
38
39      s = np.random.choice([-1, 1])
40      theta = np.random.uniform(-1, 1)
41
42      total_error = 0
43      for x, y in sorted_data_points_list:
44          if x - theta > 0:
45              sign = 1
46          else:
47              sign = -1
48          prediction = s * sign
49          if prediction != y:
50              total_error += 1
51      avg_total_error = total_error / 12
52
53      # aim: compute E_out(g)
54      v = s * 0.35
55      u = 0.5 - v
56      E_out = u + v * abs(theta)
57
58      # aim: store the result of in sample error and out of sample error, and their differences
59      E_in_all_experiments.append(avg_total_error)
60      E_out_all_experiments.append(E_out)
61      difference_list.append(E_out - avg_total_error)
62
63  median_difference = np.median(difference_list)
```