Noise, error & co.

1. Implement polynomial regression, ie. linear regression with polynomial features. Use polynomials of degree 0, 1, 2 and 5.

   a. Find the parameters that generated the dataset in `poly_s.npz`. Print training and test MSEs for each polynomial. What do you think, what was the degree of the original polynomial (the target function), and why (visually and based on the test error)?[1]

   b. Find the parameters that generated the dataset in `poly_d.npz`. Print training and test MSEs for each polynomial. What do you think, what was the degree of the original polynomial, and why (visually and based on the test error)?

   c. Plot the polynomials defined by the parameters found in a) and b) along with the training and test samples. To plot a polynomial, simply use `np.linspace` on the $[-5, 5]$ interval, calculate the values for each point, then use `plt.plot` with a continuous linestyle (which is the default, by the way).

   d. Calculate the E_{out} analytically, assuming the inputs are uniformly distributed on $[-5; 5]$. The generating polynomial

      - in `poly_s` is $p(x) = 3 + 0.5x + 5x^2$ (plus some zero-mean noise, but you don't have to deal with that here),
      - in `poly_d` is $p(x) = 3 + 0.5x + 5x^2 - 0.0001x^7 + 0.00004x^8$ (no noise).

2. Plot the learning curve (here it means E_{in} and E_{out} versus the training dataset size) of polynomial regression when approximating a sine function. For this, generate random datasets of sizes ranging to 100 on the input interval $[-5\pi, +5\pi]$, train a polynomial model (similarly to Exercise 1) on this dataset, and measure the in sample error.

   Then estimate the out of sample error by taking 500 evenly spaced points on the interval, and measuring the difference between the output of your model and $\sin(x)$ on these data points. Average your measurements across 50 experiments (with new training data points sampled randomly from the interval). Finally, plot E_{in} and E_{out} versus the size of the dataset.

   a. So your task is basically:

      - take 500 evenly spaced points in the above interval (`np.linspace`) and apply the sine function; this will be your test dataset (for the whole exercise)
      - pick a polynomial degree d
      - for all training dataset sizes k ranging from $d + 1$ to 100, do the following:
        - sample k amount of points (uniformly randomly) from the interval (*not* from the test set), apply the sine function; this will be your training dataset for this iteration
        - fit your polynomial, measure E_{in}, E_{out}
        - do this 50 times, averaging the errors
      - plot E_{in} and E_{out} versus the dataset size (learning curves)

   b. Try this for polynomials of different degree (eg. 0, 1, 2, 5, 8). Note the minimal dataset size for each degree! Based on your experiments, which polynomial (degree) would you use if you had 20 points in your training dataset?

[1] What would you have thought before you looked the actual solution later in the Exercise? :)