Problem 2

Part A

If initial model is **high variance mode** so more training data will help lower the variance of a high variance model since there will be less overfitting if the learning algorithm is exposed to more data samples. But more data

Bias, is defined as

$$Bias\left[\hat{f}(x)\right] = E\left[\hat{f}(x)\right] - f(x)$$

So would not be affected by increasing the training set size.

In the other hand, if initial model is **high bias model** so more training data doesn't help lower the bias of the high bias model

Part B

1. **Hold-out (data)**
   Rather than using all of our data for training, we can simply split our dataset into two sets: training and testing. A common split ratio is 80% for training and 20% for testing. We train our model until it performs well not only on the training set but also for the testing set. This indicates good generalization capability since the testing set represents unseen data that were not used for training. However, this approach would require a sufficiently large dataset to train on even after splitting.

2. **Cross-validation (data)**
   We can split our dataset into k groups (k-fold cross-validation). We let one of the groups to be the testing set (please see hold-out explanation) and the others as the training set, and repeat this process until each individual group has been used as the testing set (e.g., k repeats). Unlike hold-out, cross-validation allows all data to be eventually used for training but is also more computationally expensive than hold-out.

3. **Data augmentation (data)**
   A larger dataset would reduce overfitting. If we cannot gather more data and are constrained to the data we have in our current dataset, we can apply data augmentation to artificially increase the size of our dataset. For example, if we are training for an image classification task, we can perform various image transformations to our image dataset (e.g., flipping, rotating, rescaling, shifting).

4. **Feature selection (data)**

   If we have only a limited amount of training samples, each with a large number of features, we should only select the most important features for training so that our model doesn't need to learn for so many features and eventually overfit. We can simply test out different features, train individual models for these features, and evaluate generalization capabilities, or use one of the various widely used feature selection methods.

5. **L1 / L2 regularization**

   Regularization is a technique to constrain our network from learning a model that is too complex, which may therefore overfit. In L1 or L2 regularization, we can add a penalty term on the cost function to push the estimated coefficients towards zero (and not take more extreme values). L2 regularization allows weights to decay towards zero but not to zero, while L1 regularization allows weights to decay to zero.