# Chapter 3: Overfitting and Underfitting

In the context of machine learning, *overfitting* and *underfitting* are two common problems related to how well a model is able to generalize from its training data to unseen, new data. These issues are critical to understand how well a model performs and how it can be improved further.

## 3.1. Overfitting

*Overfitting* occurs when **a model learns the training data too well, capturing noise and random fluctuations rather than the intended patterns**. This typically results in a model that performs exceptionally well on the training data but poorly (means does not generalize well) on new, unseen data. Overfitting is akin to memorizing the training data rather than learning the underlying structure.

### 3.1.1. Causes of Overfitting

In summary, the possible causes of overfitting are:

- **Complex Models**: Using models that are too complex relative to the amount of training data (e.g., deep neural networks, high-degree polynomial regression)
- **Insufficient Training Data**: Having a small training dataset, which can cause the model to pick up noise as if it were a true signal. More the amount of training data, more the model will be able to differentiate between noise and true patterns
- **Lack of Regularization**: Not applying regularization techniques which penalizes over complex models.

**Constraining a model to make it simpler and reduce the risk of overfitting is called** ***regularization*****.**

The amount of regularization to apply during learning can be controlled by a *hyperparameter*, a parameter of a learning algorithm (not of the model). That means, it is not affected by the learning algorithm itself; it must be set prior to training and remains constant during training. If the regularization hyperparameter is set to a very large value, the learning algorithm will almost certainly not overfit the training data, but it will be less likely to find a good solution. Tuning hyperparameters is an important part of building a Machine Learning system.

### 3.1.2. Signs of Overfitting

- **High Variance**: The model has high accuracy on the training set but low accuracy on the validation/test set.
- **High Complexity**: The model has many parameters and layers (compared to the size of the training data) or uses high-degree polynomials.

### 3.1.3. Solutions for Overfitting

- **Simplify the Model**: Use a simpler model with fewer parameters compared to the amount of training data.
- **Regularization**: Apply techniques such as L1 (Lasso) or L2 (Ridge) or L1 and L2 combined (Elastic Net) regularization to constrain model complexity.
- **Cross-Validation**: Use cross-validation techniques (e.g., k-fold cross-validation) to ensure the model generalizes well to unseen data.
- **More Data**: Increase the size of the training dataset to provide more examples of the underlying patterns and help the model to differentiate noise from the patterns.
- **Dropout**: Use dropout layers in neural networks to prevent units from co-adapting too much.

## 3.2. Underfitting

*Underfitting* is exactly the opposite of overfitting. It occurs when **a model is too simple to capture the underlying patterns in the data**. It fails to learn the training data effectively and also performs poorly on unseen data. Underfitting is like trying to fit a straight line to a dataset that has a more complex shape.

### 3.2.1. Causes of Underfitting

- **Simple Models**: Using models that are too simple relative to the complexity of the data may not be able to completely capture the underlying complex patterns (e.g., linear regression for non-linear data).
- **Insufficient Training Time**: Not training the model long enough to learn the patterns.
- **Over-regularization**: Applying too much regularization, which constrains the model too tightly.

### 3.2.2. Signs of Overfitting

- **High Bias**: The model has poor performance on both the training set and the validation/test set.
- **Poor Accuracy**: Low accuracy across all datasets due to a lack of model capacity.

### 3.2.3. Solutions for Underfitting

The possible solutions to underfitting are:

- **Increase Model Complexity**: Use more complex models with more parameters or layers.
- **Reduce Regularization**: Decrease regularization hyperparameters to allow the model more freedom to fit the data (reduce the constraints on the model).
- **Feature Engineering**: Add more relevant features or transform existing features to better capture the underlying patterns.

- **Train Longer**: Increase training time or iterations, especially in iterative models like neural networks (to allow more time to model to learn from the data)

**Our model needs to be neither too simple (in which case it will underfit) nor too complex (in which case it will overfit). It should be a balance of both.**

## Bias-Variance Trade-off

The problems of overfitting and underfitting are closely related to the bias-variance trade-off:

- **Bias**: Bias is the error due to overly simplistic assumptions in the learning algorithm. It quantifies how far the expected prediction is from the true value in a specific dataset. If the model is oversimplified, it will be unable to capture the underlying pattern. Then the predicted value would be far from the ground truth, resulting in more bias. Therefore, **high bias can cause underfitting**.

  $$Bias\left(\hat{f}_{\hat{\theta}}(x)\right) = \mathbb{E}\left[\hat{f}_{\hat{\theta}}(x)\right] - f_{\theta^*}(x)$$

  Here, $f_{\theta^*}(x)$ is the true value corresponding to the true parameter $\theta^*$, and $\mathbb{E}\left[\hat{f}_{\hat{\theta}}(x)\right]$ is the expected prediction corresponding to the estimated parameter $\hat{\theta}$.

- **Variance**: Variance is the error due to excessive sensitivity to small fluctuations in the training data. It measures the amount by which $\hat{f}_{\hat{\theta}}(x)$ would change if we estimate it using a different training dataset. Therefore, it captures the sensitivity of the estimator to the random fluctuation of the training dataset. If the model's performance is tested on different datasets, the closer the prediction, the lesser the variance. A high variance indicates the model is too complex and is too sensitive to the dataset it was trained on, i.e., it captures the noise in the training dataset rather than learning the pattern, resulting in poor generalization to new data. Therefore, **high variance can cause overfitting**.

  $$Var\left(\hat{f}_{\hat{\theta}}(x)\right) = \mathbb{E}\left[\left(\hat{f}_{\hat{\theta}}(x) - \mathbb{E}\left[\hat{f}_{\hat{\theta}}(x)\right]\right)^2\right]$$

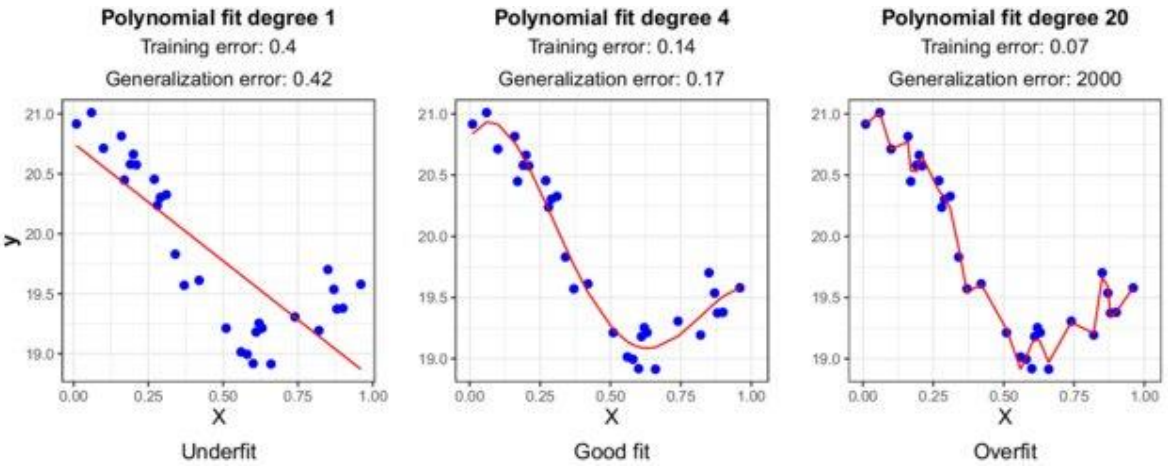
If we aim to reduce the variance, we will increase bias and vice versa. The goal is to find a trade-off between bias and variance that minimizes the total error. Complex models with low bias and high variance can capture the training data very well but may not generalize to the new data. On the contrary, simple models with high bias and low variance may not capture the training data very well but can generalize better to new data.

$$Generalization\ Error\ =\ Bias^2 + Variance + Irreducible\ Error$$

## Summary

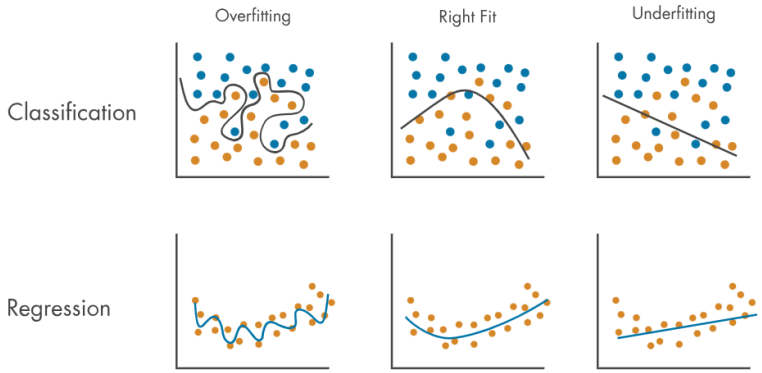
- **Overfitting**: Model is too complex. Fits training data too closely. Very sensitive to small fluctuations in the training set (low bias).

- **Underfitting**: Model is too simple. Fails to capture underlying patterns and poor performance on both training and test data (high bias).




Source: https://www.researchgate.net/publication/339680577_An_Introduction_to_Machine_Learning/figures?lo=1

The above image shows what underfitting and overfitting looks like in polynomial regression.




Source: https://www.mathworks.com/discovery/overfitting.html

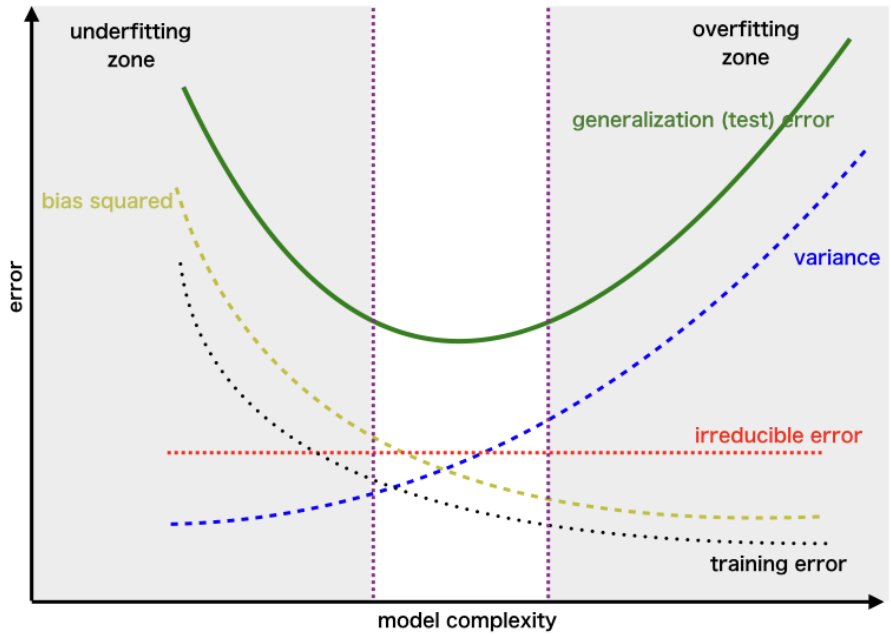
This is how overfitting and underfitting looks like for regression and classification problems.

Finding the right balance between overfitting and underfitting involves selecting an appropriate model complexity and using techniques such as regularization and cross-validation to improve generalization to unseen data.

Overfitting and underfitting also impacts the training and the generalization error. As we can see in the below example plot:

- Underfitting zone has high bias
- Overfitting zone has high variance
- *Training error* is the error rate of the model on the training data. It indicates how well the model has learned the training dataset. In the underfitting zone, simple model cannot fit the training data well, so training error is high. As complexity increases, the model learns the underlying pattern, hence reducing training error.

- *Generalization error* (sometimes, also referred as validation error) is the error rate of the model on new, unseen data. It indicates the model's ability to generalize beyond the training dataset.
    - In the underfitting zone, the generalization error is high because of high bias.
    - In the overfitting zone, the generalization error is high because of high variance
    - In the balanced region, the generalization error is minimum.




In general, this is how training and generalization error is impacted by the overfitting and underfitting:

| ***Error*** | **Overfitting** | **Right Fit** | **Underfitting** |
|---|---|---|---|
| ***Training*** | Low | Low | High |
| ***Test*** | High | Low | High |