Machine Learning 101

Rajdeep Chatterjee, Ph.D.
Amygdala AI, Bhubaneswar, India *

January 2025

Regularization

1 Bias and Variance in Linear Models

Bias and variance are fundamental concepts in machine learning, describing the sources of error in predictive models. Together, they form the **bias-variance tradeoff**, which impacts the model's generalization to unseen data.

1.1 Bias

Bias measures the error introduced by approximating a real-world problem, which may be complex, using a simplified model. High bias indicates that the model is too simplistic and fails to capture the underlying patterns in the data.

Mathematical Definition: Bias is defined as the difference between the expected prediction of the model and the true value:

$$\text{Bias}^2 = (E[f(x)] - f_{\text{true}}(x))^2$$

where:

- $f(x)$: Predicted output by the model.
- $f_{\text{true}}(x)$: True function generating the data.

Example: Using a linear regression model to fit a highly nonlinear dataset results in high bias. The model oversimplifies the relationship, failing to capture critical data trends.

*Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* `http://amygdalaai.org/`.

1.2 Variance

Variance measures the sensitivity of the model to fluctuations in the training data. High variance indicates that the model is overly complex and captures noise in the data, leading to overfitting.

Mathematical Definition: Variance is defined as the expected variability of the model's predictions around its mean prediction:

$$\text{Variance} = E\left[(f(x) - E[f(x)])^2\right]$$

Example: A high-degree polynomial regression model fits every detail and noise in the training dataset, resulting in poor generalization to unseen data.

1.3 Bias-Variance Tradeoff

The total error (mean squared error) in a model can be decomposed into three components:

$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- **High Bias:** Leads to underfitting, where the model cannot capture the underlying patterns.
- **High Variance:** Leads to overfitting, where the model captures noise along with the true signal.
- **Irreducible Error:** Noise inherent in the data that cannot be eliminated.

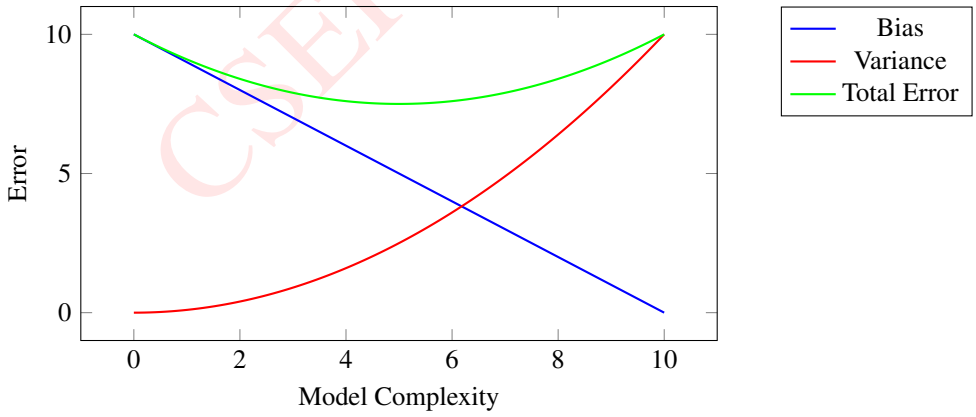

Figure 1: Bias-Variance Tradeoff as a Function of Model Complexity

1.4 Examples of Bias and Variance

1. **High Bias (Underfitting):** Using a linear regression model to predict housing prices, where the relationship between features and target is nonlinear, results in systematic error (high bias).
2. **High Variance (Overfitting):** Using a polynomial regression of degree 10 for a small dataset fits every noise in the data. While the model performs well on training data, it generalizes poorly to new data.
3. **Optimal Tradeoff:** A polynomial regression of degree 3 fits the data well, balancing bias and variance. It captures the overall trend without overfitting to noise.

2 Overfitting and Underfitting

In machine learning, the concepts of overfitting and underfitting are critical for understanding model performance.

2.1 Overfitting

Overfitting occurs when a model learns not only the underlying pattern but also the noise in the training data. As a result, the model performs well on training data but poorly on unseen test data.

- **Characteristics:** High training accuracy but low test accuracy.
- **Cause:** Model complexity is too high, allowing it to memorize the training data.
- **Example:** A high-degree polynomial fits training data perfectly but fails to generalize.

2.2 Underfitting

Underfitting occurs when a model fails to capture the underlying pattern in the data. The model performs poorly on both training and test data.

- **Characteristics:** Low training and test accuracy.
- **Cause:** Model complexity is too low to represent the data effectively.
- **Example:** A low-degree polynomial fails to capture the curvature of the data.

2.3 Illustration

3 Occam's Razor

Occam's Razor is a principle suggesting that the simplest explanation is often the best. In machine learning, this implies preferring simpler models to avoid overfitting.

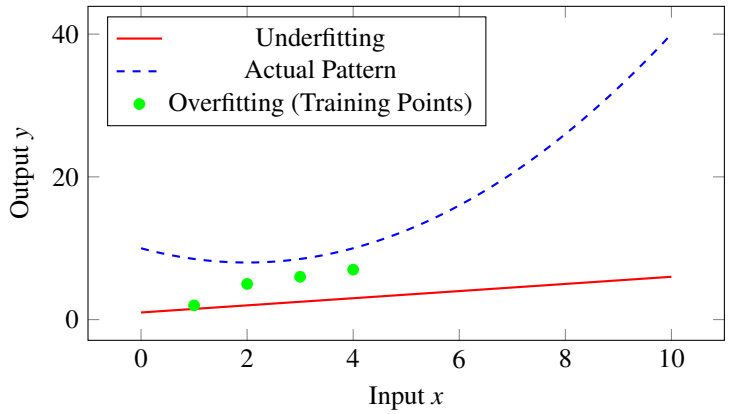

Figure 2: Visualization of underfitting, actual pattern, and overfitting.

3.1 Application in Machine Learning

- A simpler model is less likely to overfit but may underfit if it is too simple.
- A complex model may overfit the training data but fail to generalize.

4 Bias-Variance Tradeoff

The bias-variance tradeoff explains the balance between bias (error due to incorrect assumptions) and variance (error due to sensitivity to small changes in training data).

4.1 Key Concepts

- **Bias:** Error from overly simplistic assumptions.
- **Variance:** Error from excessive sensitivity to training data.
- **Tradeoff:** Increasing model complexity reduces bias but increases variance, and vice versa.

4.2 Mathematical Formulation

The expected error can be expressed as:

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error} \tag{1}$$

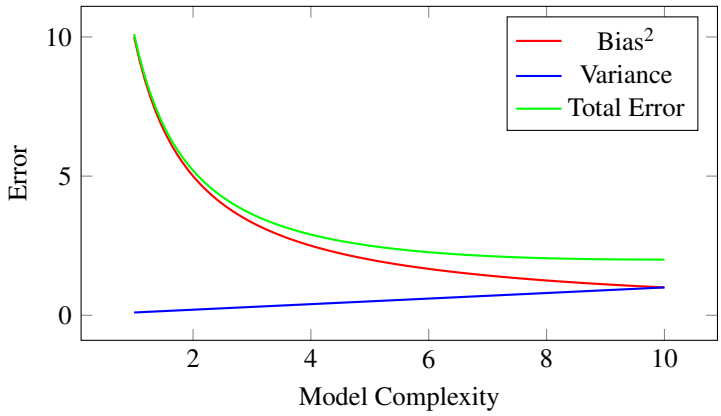

Figure 3: Bias-Variance Tradeoff as model complexity increases.

4.3 Illustration of Bias-Variance Tradeoff

5 Examples

5.1 Overfitting Example: High-Degree Polynomial

Consider fitting a 10th-degree polynomial to 10 data points. The model fits training data perfectly but fails to generalize.

5.2 Underfitting Example: Linear Model

Consider fitting a straight line to a quadratic data pattern. The model fails to capture the curvature.

6 Preventing Overfitting and Underfitting

To achieve optimal performance in machine learning models, it is essential to prevent both overfitting and underfitting. Here, we discuss common strategies to mitigate these problems.

6.1 Regularization

Regularization is a technique to reduce overfitting by penalizing large model coefficients. By adding a penalty term to the loss function, regularization discourages the model from relying too heavily on any one feature or introducing excessive complexity.

6.1.1 Types of Regularization

Regularization methods differ in how they penalize the model coefficients. Below are three common types:

- **L1 Regularization (Lasso):** L1 regularization adds a penalty proportional to the absolute value of the coefficients. This can shrink some coefficients to zero, effectively removing irrelevant features and performing feature selection. The loss function is:

$$\text{Loss} = \text{MSE} + \lambda \sum_{i=1}^{n} |w_i|$$

  Benefits:

  - Sparse solutions with many zero coefficients.
  - Useful for high-dimensional datasets.

  Drawback:

  - Lasso struggles when features are highly correlated.

- **L2 Regularization (Ridge):** L2 regularization adds a penalty proportional to the square of the coefficients, resulting in smaller, non-zero coefficients. The loss function is:

$$\text{Loss} = \text{MSE} + \lambda \sum_{i=1}^{n} w_i^2$$

  Benefits:

  - Distributes weight across correlated features.
  - Reduces model sensitivity to noise.

  Drawback:

  - Does not perform feature selection, as coefficients shrink but rarely reach zero.

- **Elastic Net:** Elastic Net combines L1 and L2 regularization, addressing the limitations of each. The loss function is:

$$\text{Loss} = \text{MSE} + \lambda_1 \sum_{i=1}^{n} |w_i| + \lambda_2 \sum_{i=1}^{n} w_i^2$$

  Benefits:

  - Handles correlated features better than Lasso.
  - Offers a balance between sparsity and distributed weights.

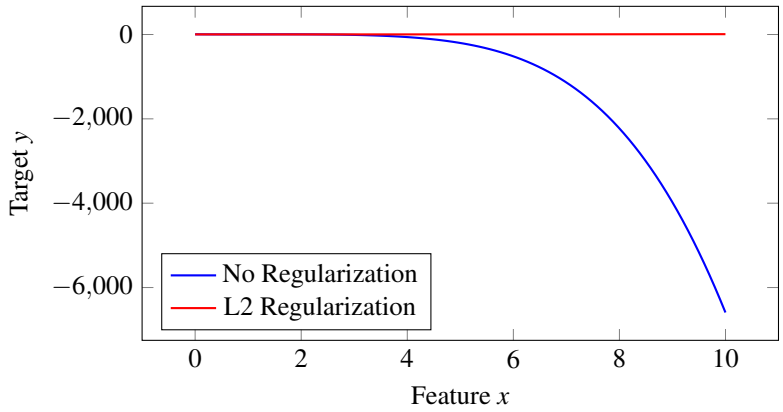

Figure 4: Impact of Regularization in Polynomial Fitting

6.1.2 Example: Regularization in Polynomial Fitting

Consider fitting a polynomial regression model to noisy data. Without regularization, a high-degree polynomial fits the noise, leading to overfitting. By applying L2 regularization, the model penalizes large coefficients associated with high-degree terms, resulting in a smoother and more generalized fit.

6.1.3 Choosing the Regularization Strength (λ)

The regularization strength λ controls the tradeoff between the model's fit to the training data and the penalty for large coefficients:

- A small λ results in minimal regularization, increasing the risk of overfitting.
- A large λ overly penalizes coefficients, leading to underfitting.

The optimal λ is typically selected using cross-validation. The validation error is plotted as a function of λ, and the value minimizing the error is chosen.

6.2 Cross-Validation

Cross-validation is a technique to evaluate model performance on unseen data by partitioning the dataset into training and validation subsets. It helps tune hyperparameters and select the best model.

6.2.1 K-Fold Cross-Validation:

The dataset is split into K subsets (folds). Each fold is used as a validation set once, and the model is trained on the remaining $K-1$ folds. The final performance is the average validation error.

$$\text{CV Error} = \frac{1}{K}\sum_{k=1}^{K}\text{Error}_k$$

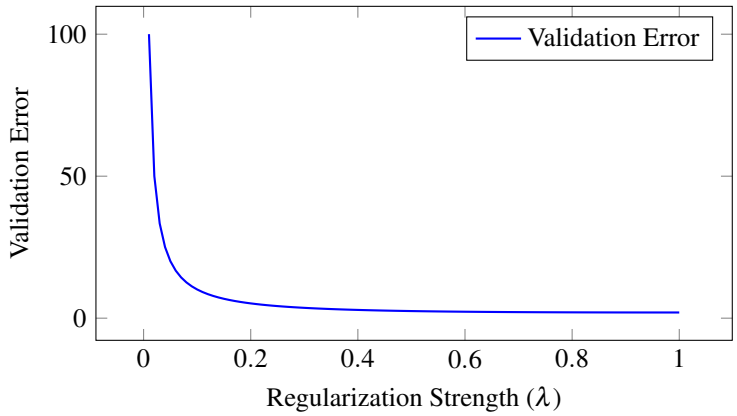

Figure 5: Selecting λ using Cross-Validation

6.2.2 Benefits:

- Ensures model generalizes well to unseen data.
- Reduces overfitting by using multiple validation sets.

6.3 Feature Selection

Irrelevant or redundant features can introduce noise, leading to overfitting. Feature selection helps retain only the most important features.

6.3.1 Methods for Feature Selection:

- **Filter Methods:** Use statistical tests (e.g., correlation or mutual information) to rank features.
- **Wrapper Methods:** Use machine learning models to evaluate subsets of features.
- **Embedded Methods:** Perform feature selection as part of the model training (e.g., L1 regularization).

6.3.2 Example:

In a dataset with hundreds of features, most might be irrelevant. Feature selection techniques like Recursive Feature Elimination (RFE) can identify a smaller subset that improves model performance.

6.4 Model Selection

Choosing a model that matches the data complexity is crucial to avoid overfitting or underfitting. Simpler models are less prone to overfitting but may underfit, while complex models may overfit the data.

6.4.1 Steps for Model Selection:

- Start with a simple model (e.g., linear regression).
- Gradually increase complexity (e.g., polynomial regression or neural networks).
- Use validation metrics to assess model performance.

6.4.2 Example:

If the data shows a quadratic trend, a linear model will underfit, while a high-degree polynomial will overfit. A quadratic model strikes the right balance.

6.5 Comparison of Strategies

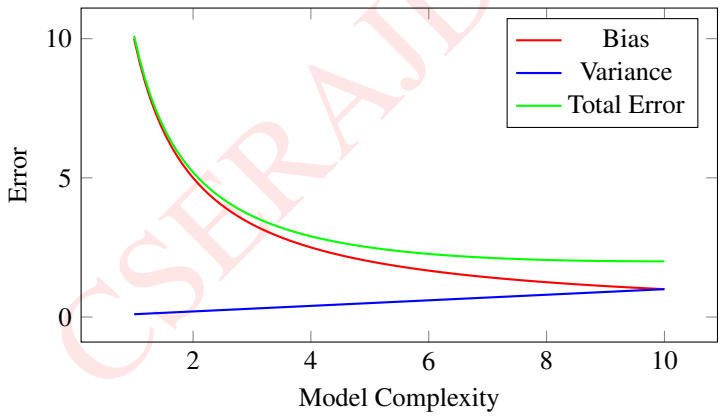

Figure 6: The tradeoff between bias, variance, and total error as model complexity increases.