CAP 4630 – Bias Vs Variance

Instructor: Aakash Kumar

University of Central Florida

Introduction to Model Performance

- Understanding Model Performance:
  - When we talk about a Machine Learning model, we refer to how well it performs in terms of accuracy and prediction errors.
  - Our goal is to design a model that generalizes well to new, unseen data, not just to the data it was trained on.
- Key Concept: Generalization:
  - A model is said to be good if it can generalize to any new input data from the problem domain.
  - Generalization helps in making accurate predictions about future data that the model has never encountered.

Overfitting and Underfitting

- Evaluating Generalization:
  - To evaluate how well a model generalizes, we compare its performance on unseen test data.
- Challenges in Generalization:
  - Overfitting: When a model performs well on training data but poorly on unseen data.
  - Underfitting: When a model performs poorly on both training and unseen data.
- Impact: Both overfitting and underfitting are major causes of poor performance in machine learning algorithms.

Bias vs Variance

Bias in Machine Learning

- What is Bias?
  - Bias refers to the error introduced by overly simplistic assumptions made by the learning algorithm.
  - These assumptions make the model easier to understand but may prevent it from capturing the full complexity of the data.
- Impact of High Bias:
  - A model with high bias is too simplified, failing to represent the relationship between input and output accurately.
  - High bias typically leads to underfitting, where the model performs poorly on both training and test data.

Consequences of High Bias

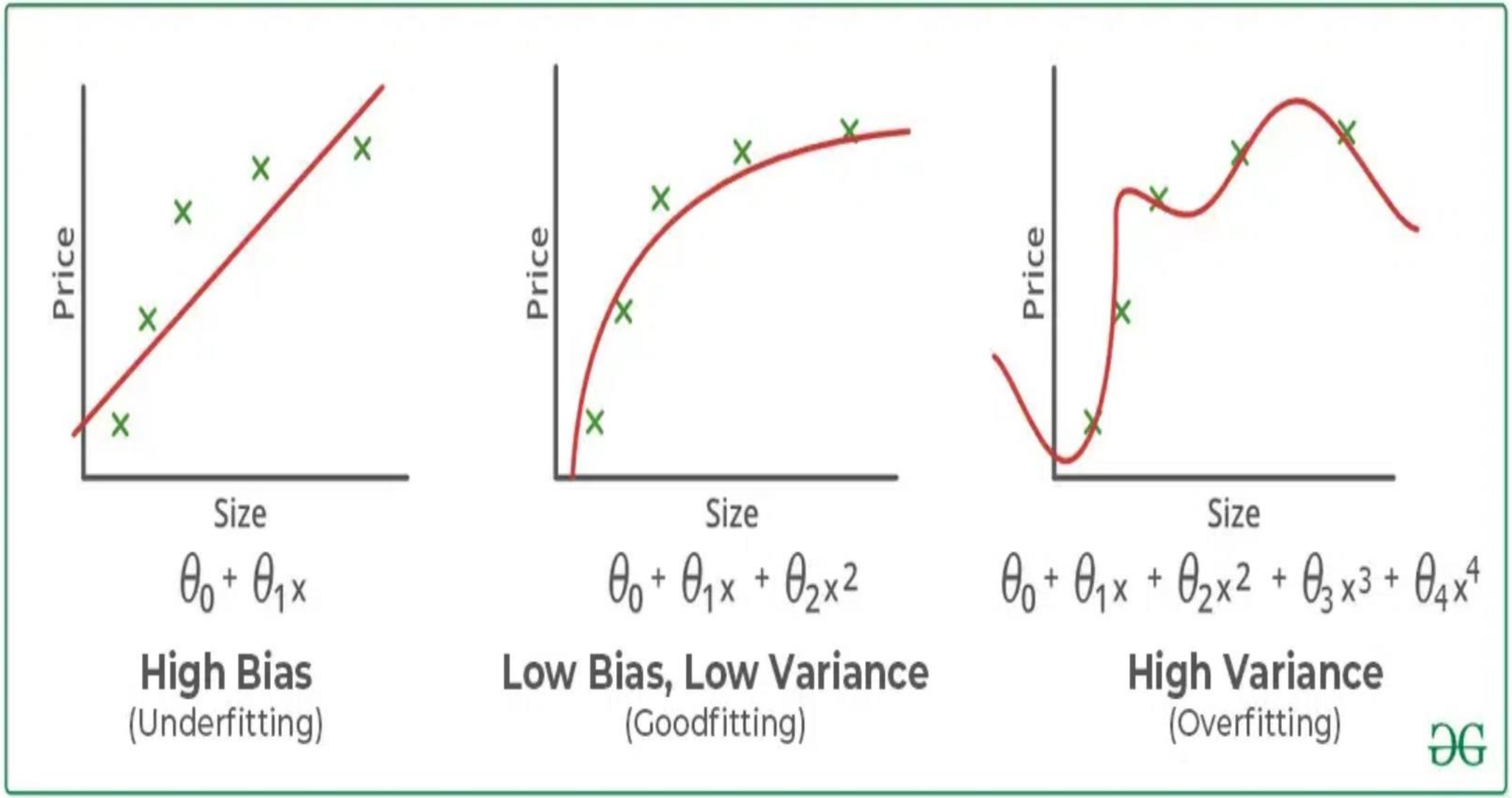
- Symptoms of High Bias:
  - Poor performance on both training and testing datasets.
  - The model is unable to capture the underlying patterns in the data.
- Example of High Bias:
  - A linear model applied to data that requires a more complex representation (e.g., nonlinear patterns).
- Real-World Implication:
  - High bias often indicates that the model needs to be more complex or have additional features to better represent the data.

Understanding Variance in Machine Learning

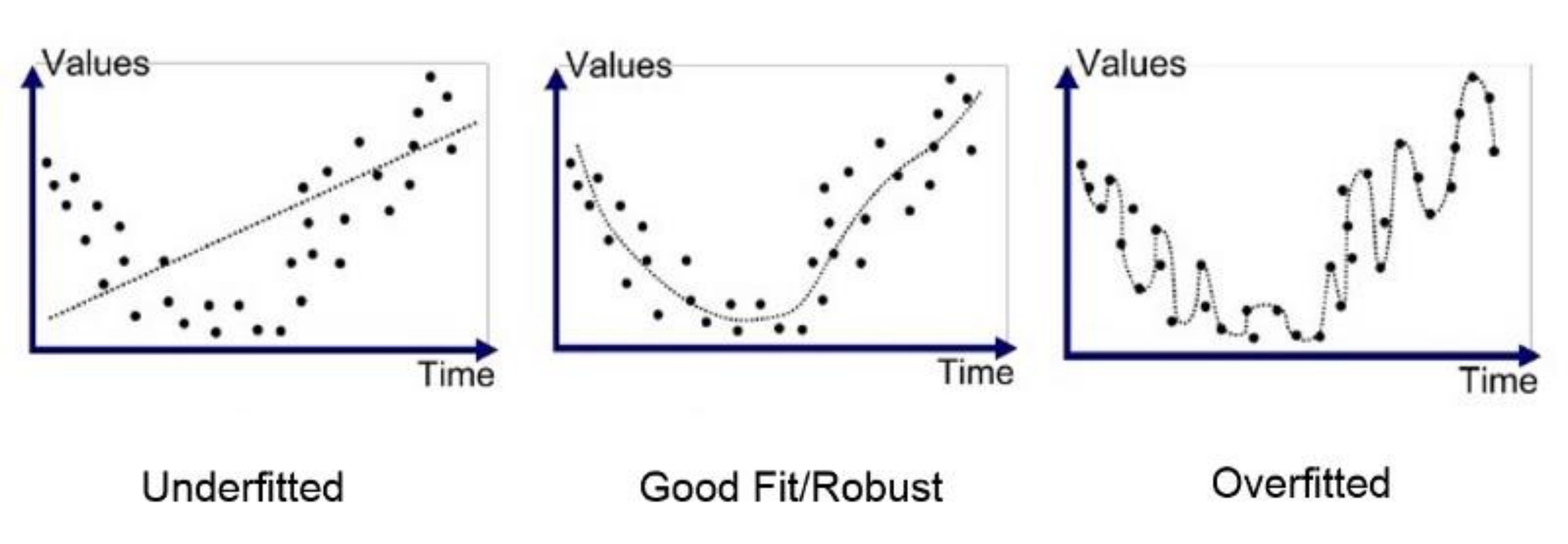
- What is Variance?
  - Variance refers to the model's sensitivity to small changes or fluctuations in the training data.
  - It is the variability in the model's predictions for different subsets or instances of the training data.
- Impact of High Variance:
  - High variance models are typically very complex and capture random noise in the training data.
  - Such models tend to perform well on the training data but fail to generalize to unseen data, leading to overfitting.

Consequences of High Variance

- Symptoms of High Variance:
  - The model performs very well on the training data but poorly on the testing data.
  - High variance means the model is too sensitive to noise and fluctuations in the training set, rather than capturing the true underlying patterns.
- Example of High Variance:
  - A highly complex decision tree model that fits perfectly to the training data but fails on test data.
- Real-World Implication:
  - High variance indicates that the model may need to be simplified by pruning or regularizing to prevent overfitting and improve generalization.



Bias and Variance


Values
Time
Values
Time
Values
Time
Underfitted
Good Fit/Robust
Overfitted

Underfitting

What is Underfitting?

- Definition: Underfitting occurs when a machine learning model is too simple to capture the underlying patterns in the data.
  - This leads to poor performance both on the training and testing datasets, as the model fails to learn from the data.
- Characteristics:
  - The model exhibits high bias and low variance.
  - It performs poorly on both known (training) data and unseen (test) data, making inaccurate predictions.

Causes of Underfitting

- Model is too simple: The model lacks complexity to capture data relationships (e.g., using a linear model for nonlinear data).
- Inadequate features: Input features do not adequately represent the factors influencing the target variable.
- Small training dataset: Insufficient data may prevent the model from learning key patterns.
- Excessive regularization: Too much regularization restricts the model's ability to learn effectively.

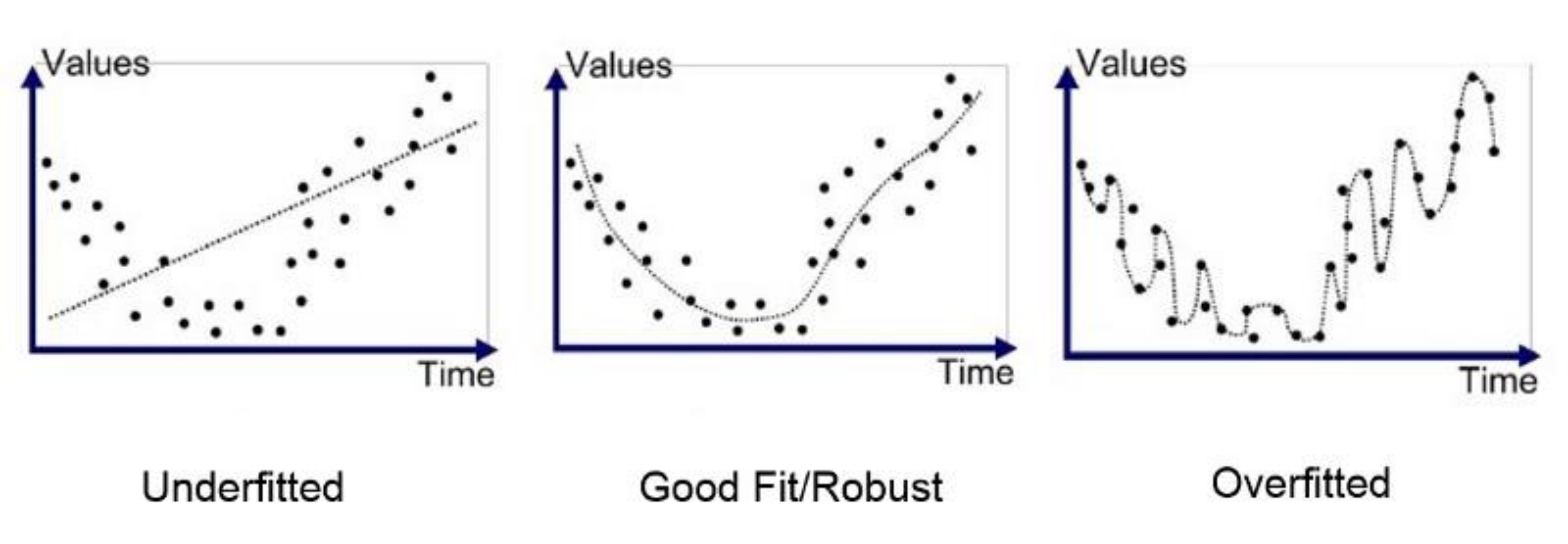
How to Address Underfitting

- Increase model complexity: Use more advanced algorithms to capture data complexities.
- Add more features: Perform feature engineering to improve representation.
- Remove noise: Clean the dataset to enhance model accuracy.
- Train longer: Increase the number of epochs or training time to allow the model to better learn from the data.

Overfitting

What is Overfitting?

- Definition: Overfitting occurs when a model learns not only the underlying patterns in the training data but also the noise and random fluctuations, causing poor generalization to unseen data.
- Key Symptoms:
  - Low bias, but high variance.
  - Excellent performance on training data, but poor performance on test data.

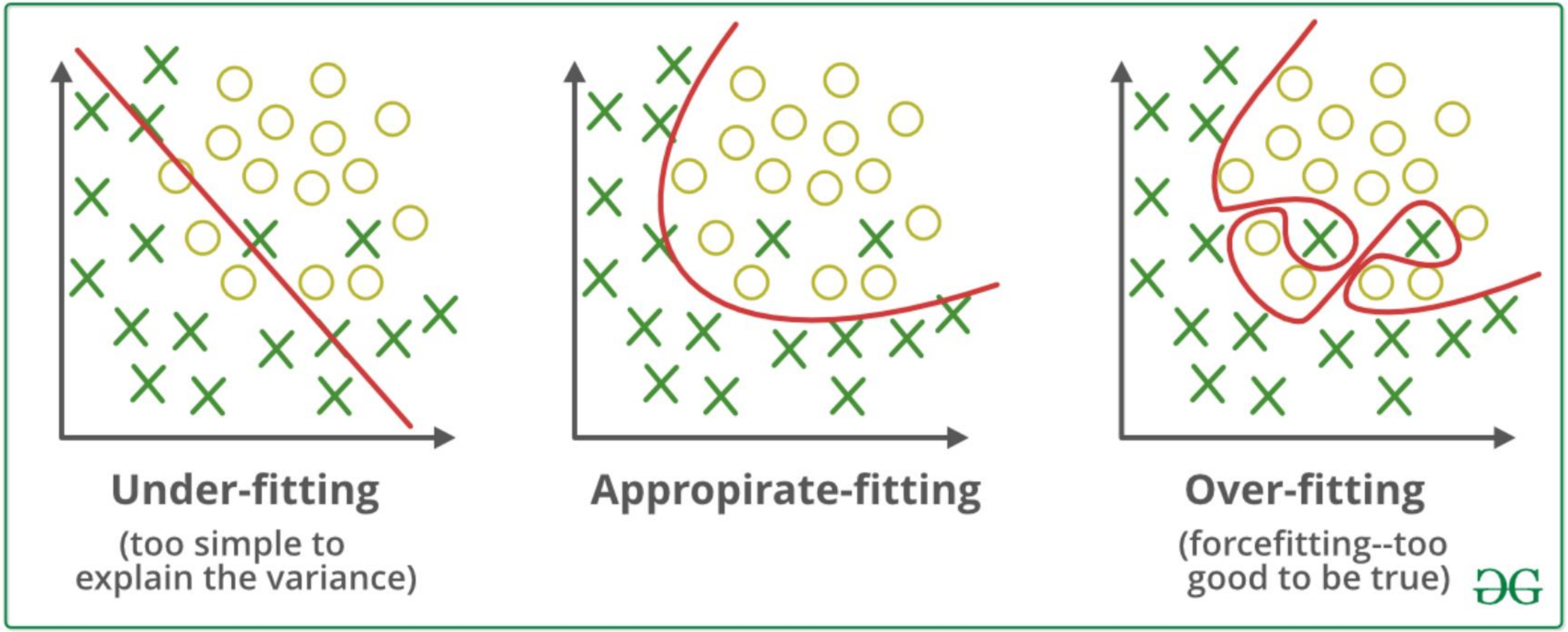

Values
Time
Underfitted
Values
Time
Good Fit/Robust
Values
Time
Overfitted

Causes of Overfitting

- Model complexity: The model is too complex, capturing noise and irrelevant patterns in the training data.
- Insufficient data: Small training datasets can lead to models that overfit due to learning irrelevant details.
- Excessive training: Training for too many epochs can cause the model to fit the data too closely

Techniques to Prevent Overfitting

- **Increase training data**: Provide more examples to help the model learn better generalizations.
- **Reduce model complexity:** Use simpler models or limit the parameters (e.g., pruning decision trees, reducing the number of layers in a neural network).
- **Regularization techniques:**
  - Lasso (L1) and Ridge (L2) regularization.
- **Early stopping:** Stop training when the performance on validation data begins to degrade.
- **Dropout (for neural networks):** Randomly drop neurons during training to avoid over-reliance on specific features.



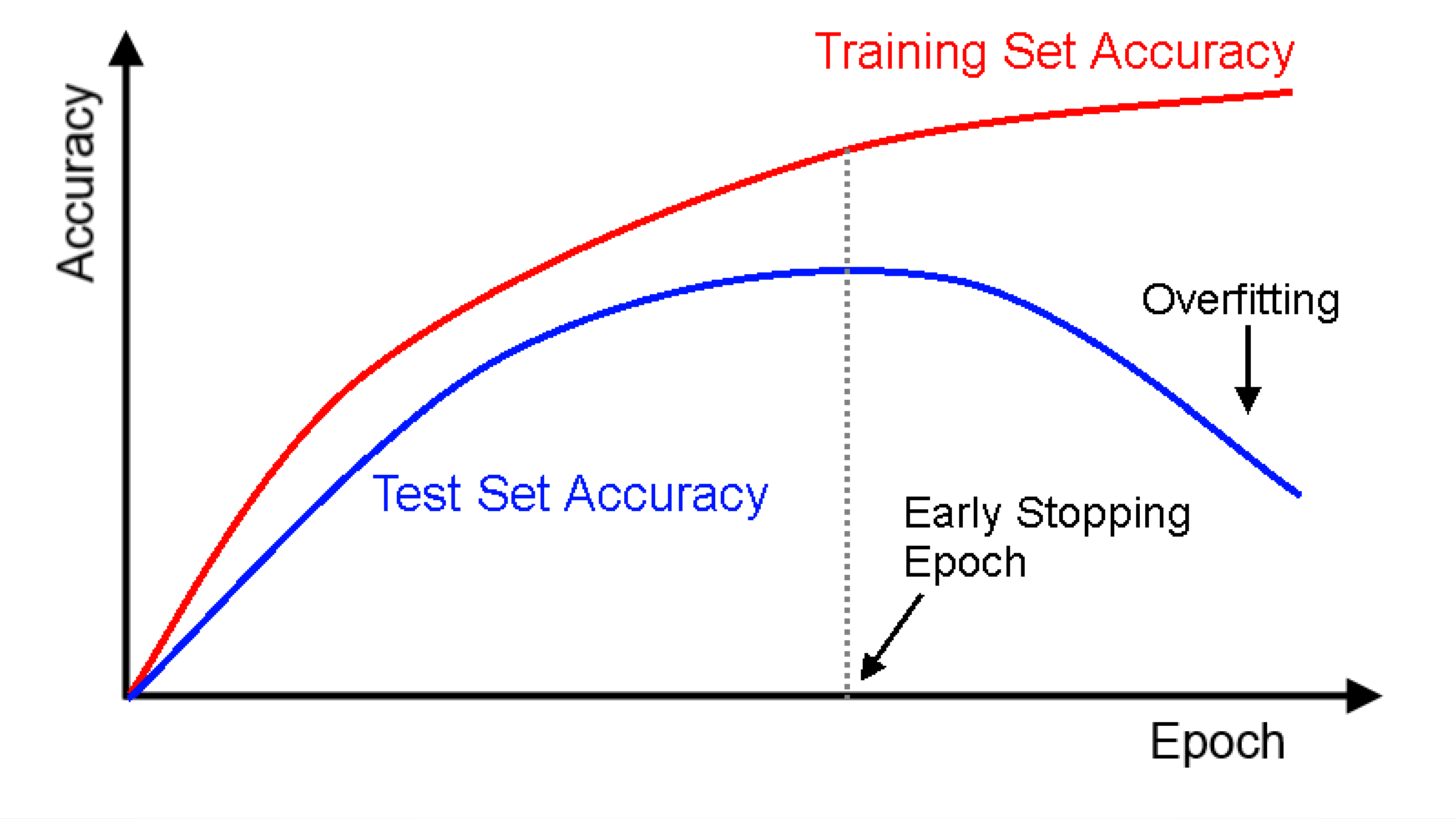
Underfitting and Overfitting

Early Stopping

Early Stopping - Concept

- What is Early Stopping?
  - During training, a model (especially a large neural network) can reach a point where it stops generalizing and begins overfitting, learning the noise in the training data.
  - Early Stopping helps to prevent this by halting training once performance on validation data starts to degrade.
- Solution:
  - Stop training as soon as the generalization error (validation error) starts increasing.
  - This prevents overfitting and ensures better performance on unseen data.

Early Stopping - Visualization



Early Stopping - Implementation and Benefits

- Why Use Early Stopping?
  - Prevents Overfitting: Stops training before the model begins to memorize noise in the training data, ensuring better generalization to new, unseen data.
  - Simplicity and Effectiveness: Easy to implement in most deep learning frameworks, yet highly effective in preventing model overfitting.
  - Efficient Use of Resources: Saves time and computational power by halting training when further progress will no longer improve performance on validation data.
- How to Implement in Practice:
  - Framework Support: Many libraries, such as Keras, provide an EarlyStopping callback. This can monitor validation loss and stop training when it starts increasing, saving the model with the best performance:
  - The callback can also be configured to **save the best weights** during training, not just when early stopping occurs.

Regularizing Techniques

Regularization in Machine Learning

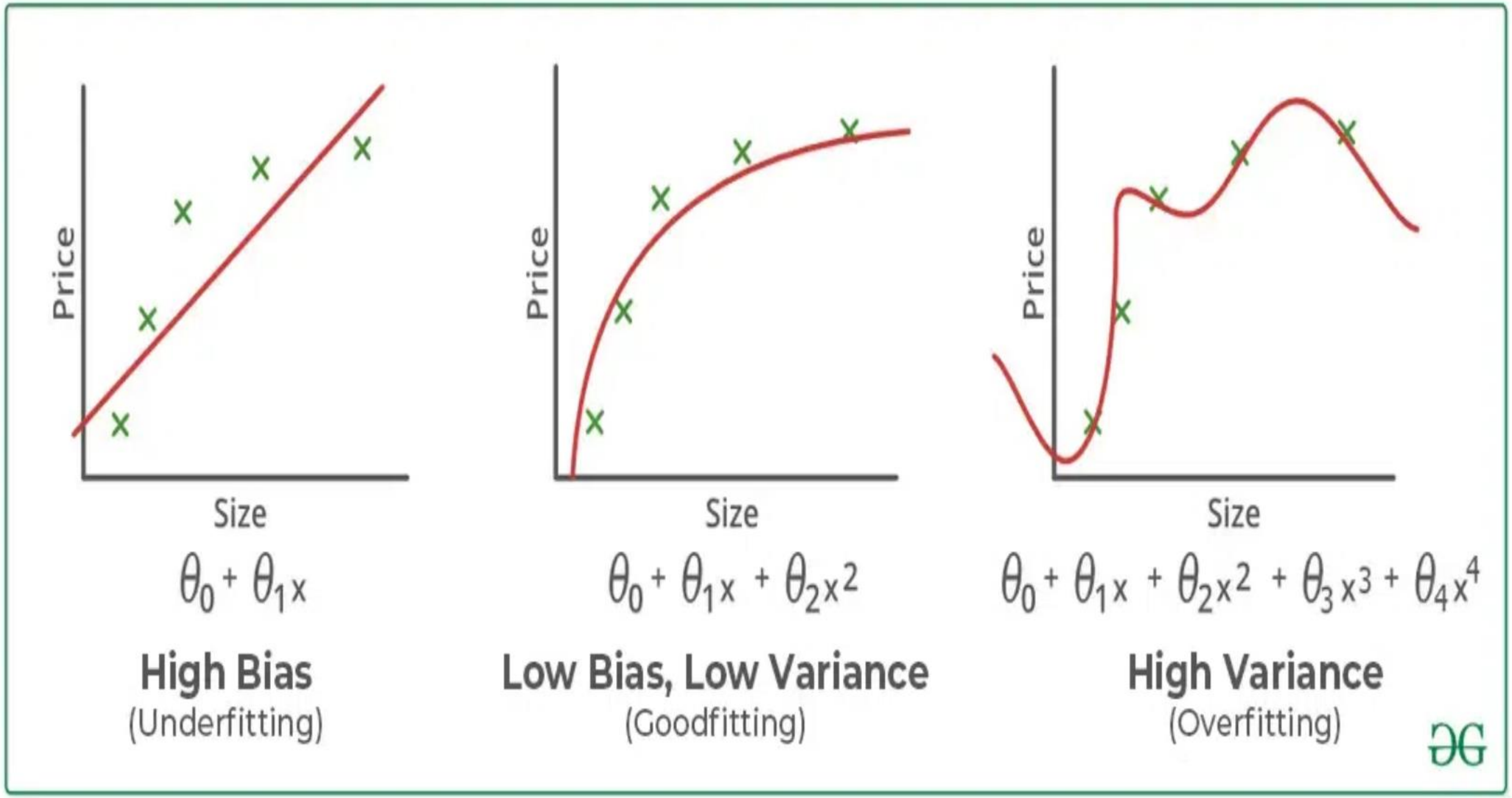
- Purpose of Regularization:
  - Regularization is used to reduce overfitting by penalizing overly complex models and encouraging simpler, more generalizable patterns.
  - It helps in striking a balance between bias and variance, ensuring that the model performs well on both the training and test datasets.

Common Regularization Techniques

- Lasso Regularization (L1): Adds a penalty proportional to the absolute value of coefficients, encouraging sparsity (i.e., many features may have zero weights).
- Ridge Regularization (L2): Adds a penalty proportional to the square of the coefficients, shrinking them, but all features are included.
- Elastic Net Regularization (L1 + L2): Combines both Lasso and Ridge regularization, balancing between shrinking and sparsity.

Introduction to Lasso Regression

- What is Lasso Regression?:
  - Lasso stands for Least Absolute Shrinkage and Selection Operator.
  - It is a type of linear regression that uses L1 regularization.
  - The goal of Lasso is not only to minimize the errors but also to shrink the coefficients of less important features to zero, effectively performing feature selection.
- When to Use Lasso Regression?:
  - Lasso is useful when we want a simple, interpretable model that selects only the most important features.



Bias and Variance

Key Idea Behind Lasso Regression

- L1 Regularization:
  - Lasso Regression adds a penalty term to the cost function, which is proportional to the absolute value of the coefficients.
- The objective function (cost function) becomes:

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m}|\theta_i|$$

How Does Lasso Perform Feature Selection?

- Feature Selection with Lasso:
  - As the regularization parameter (λ) increases, the coefficients for less important features shrink towards zero.
  - If λ is large enough, many coefficients become zero, effectively eliminating features.
  - This is why Lasso is used not just for prediction but also for selecting the most relevant features.

Impact of the Regularization Parameter λ

- Tuning the Regularization Parameter:
  - Small λ: The penalty term has little impact, leading to a model similar to ordinary least squares regression.
  - Large λ: Stronger penalty, resulting in fewer features being used in the model as more coefficients shrink to zero.
- How to Choose λ?:
  - Use cross-validation to find the optimal λ that balances bias and variance, and avoids overfitting or underfitting.

Advantages and Disadvantages of Lasso Regression

- Advantages:
  - Feature Selection: Automatically selects important features by shrinking the irrelevant ones to zero.
  - Interpretability: Results in simpler, interpretable models with only the most relevant features.
  - Avoids Overfitting: Helps to prevent overfitting by penalizing large coefficients.
- Disadvantages:
  - Bias Introduced: The model may introduce bias as coefficients are shrunk.

Introduction to Ridge Regression

- What is Ridge Regression?
  - Ridge regression is a type of linear regression that incorporates L2 regularization to prevent overfitting.
  - It is designed to improve the generalization of the model by adding a penalty for large coefficients.
- Key Feature of Ridge Regression:
  - Unlike ordinary least squares (OLS) regression, Ridge shrinks the regression coefficients by applying a penalty proportional to the squared magnitude of the coefficients.

Ridge Regression - Objective Function

- Cost Function in Ridge Regression:
  - The objective function (or cost function) in Ridge regression is modified as follows

$$\text{Cost} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{i=1}^{m}\theta_i^2$$

y_i is the actual target value.

$\hat{y}_i$ is the predicted target value.

θ_i are the coefficients (weights) of the features.

λ is the **regularization parameter** that controls the strength of the penalty.

$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ represents the residual sum of squares (RSS).

How Ridge Regression Works

- Effect of the Regularization Parameter λ:
- When λ=0:
  - Ridge regression reduces to ordinary least squares (OLS) regression, meaning there's no penalty for large coefficients.
- As λ increases:
  - The penalty for larger coefficients becomes stronger, and the coefficients shrink, preventing overfitting by reducing the influence of irrelevant or noisy features.
- Key Concept: Ridge regression trades off bias and variance to improve the model's generalization ability. A larger λ will increase bias but reduce variance, leading to better generalization on unseen data.

Advantages and Disadvantages of Ridge Regression

- Advantages:
  - Prevents Overfitting: By shrinking coefficients, Ridge helps the model generalize better, especially when there are many features or noisy data.
  - Works Well in Multicollinearity: In cases where there are high correlations among features, Ridge regression helps stabilize the model.
  - Retains All Features: Unlike Lasso, Ridge regression keeps all the features in the model, which is beneficial when all features contribute to the prediction.
- Disadvantages:
  - Less Interpretability: Since Ridge does not perform feature selection (i.e., it doesn't set coefficients to zero), it can be harder to interpret the final model.
  - Bias-Variance Tradeoff: As λ increases, the model becomes biased, and choosing an optimal λ is crucial.

Comparison of Lasso (L1) vs. Ridge (L2) Regularization

- L1 Regularization (Lasso):
  - Feature Selection: L1 regularization drives some coefficients to exactly zero, which results in a sparse model. This makes it ideal for feature selection.
  - Unimportant features get zero coefficients, effectively removing them from the model.
  - Sparsity: Leads to models with fewer non-zero coefficients, which can simplify model interpretation.
  - Example: After applying L1 regularization on a layer with 4 weights, the coefficients might look like:
  - $\theta_1 = 0.8$, $\theta_2 = 0$, $\theta_3 = 1$, $\theta 4 = 0$

Comparison of Lasso (L1) vs. Ridge (L2) Regularization

- L2 Regularization (Ridge):
  - Coefficient Shrinking: L2 regularization shrinks all coefficients towards zero, but none of the coefficients become zero.
  - This means all features remain in the model, but their influence is reduced.
  - Small Coefficients for All Features: Useful when you believe that all features contribute somewhat to the prediction, but their impact needs to be controlled.
  - Example: After applying L2 regularization on the same layer, the coefficients might look like:
  - $\theta_1=0.3,\ \theta_2=0.1,\ \theta_3=0.3,\ \theta_4=0.2$

References

- [1] https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76
- [2] Pattern Recognition and Machine Learning, M. Bishop
- [3] https://stats.stackexchange.com/questions/45643/why-l1-norm-for-sparse-models
- [4] Deep Learning, Goodfellow et. al
- [5] https://medium.com/datadriveninvestor/l1-l2-regularization-7f1b4fe948f2
- [6] Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Sergey Ioffe et al
- [7] https://towardsdatascience.com/batch-normalization-8a2e585775c9
- [8] Dropout: A Simple Way to Prevent Neural Networks from Overfitting Srivastava et al
- [9] https://machinelearningmastery.com/train-neural-networks-with-noise-to-reduce-overfitting/
- [10] Popular Ensemble Methods: An Empirical Study, Optiz et. al