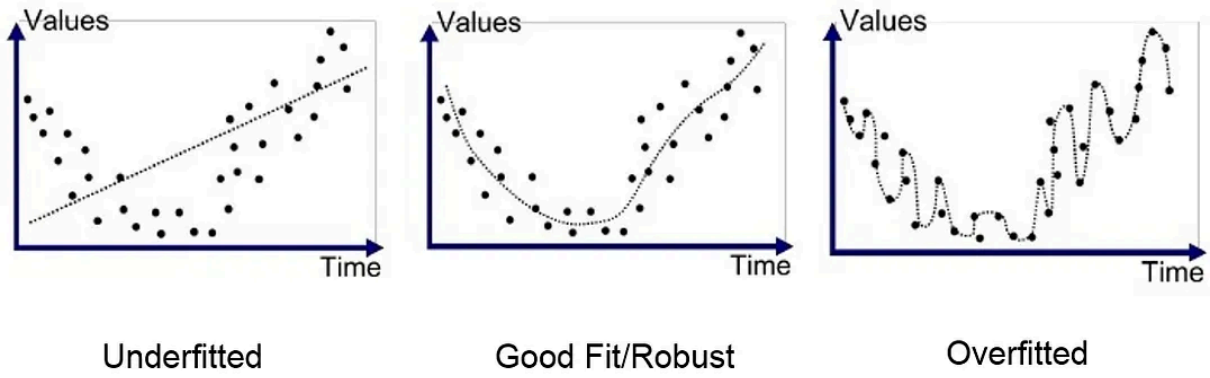
Overfitting und Underfitting im maschinellen Lernen

Diese Begriffe beschreiben typische Probleme bei der Modellanpassung an Daten. Sie zeigen, wie gut ein Modell lernt und ob es verallgemeinerbare Erkenntnisse aus den Trainingsdaten ziehen kann.



Overfitting

- **Definition**: Overfitting tritt auf, wenn ein Modell die Trainingsdaten **zu genau** lernt, einschließlich deren Rauschen, Ausreißer und unnötiger Details. Das Modell ist dann so an die Trainingsdaten angepasst, dass es schlecht auf neue, unbekannte Daten generalisiert.

- **Merkmale**:

    - **Hohe Leistung auf den Trainingsdaten**: Das Modell erzielt fast perfekte Ergebnisse im Training.
    - **Schlechte Leistung auf Testdaten**: Es kann die Muster in neuen Daten nicht verallgemeinern.
    - **Übermäßig komplexe Modelle**: Ein zu tiefes neuronales Netz oder ein Entscheidungsbaum mit vielen Knoten passt sich an spezifische Details der Trainingsdaten an, die für das eigentliche Problem irrelevant sind.

- **Beispiel**:
  Ein Entscheidungsbaum, der so tief wächst, dass er für jedes Trainingsexemplar eine eigene Regel hat, aber bei neuen Daten völlig versagt.

- **Mögliche Lösungsansätze**:

- **Regularisierung**: Begrenzen der Modellkomplexität durch Methoden wie L1- oder L2-Regularisierung.
- **Cross-Validation**: Verwenden von Validierungsdatensätzen, um frühzeitig zu erkennen, ob das Modell überfittet.
- **Mehr Trainingsdaten**: Ein größerer Datensatz hilft dem Modell, besser zu generalisieren.

Underfitting

- **Definition**: Underfitting tritt auf, wenn ein Modell zu **einfach** ist, um die zugrunde liegenden Muster in den Daten zu erfassen (oder wenn das Modell nicht zu den gegebenen Daten passt, siehe BSP unten). Es scheitert daran, sowohl die Trainingsdaten als auch neue Daten korrekt zu beschreiben.

- **Merkmale**:

  - **Niedrige Genauigkeit auf Trainingsdaten**: Das Modell erfasst nicht einmal die Muster in den Trainingsdaten.
  - **Niedrige Genauigkeit auf Testdaten**: Es generalisiert ebenfalls schlecht auf neue Daten.
  - **Zu einfache Modelle**: Modelle mit geringer Kapazität, wie eine lineare Regression für nichtlineare Daten oder ein Entscheidungsbaum mit sehr wenigen Ebenen.

- **Beispiel**:
  Eine lineare Regression wird auf Daten mit einer komplexen, nichtlinearen Beziehung angewendet. Das Modell ist nicht in der Lage, die Abhängigkeiten korrekt zu beschreiben.

- **Mögliche Lösungsansätze**:

  - **Erhöhen der Modellkomplexität**: Verwenden komplexerer Algorithmen (z. B. SVM mit nichtlinearem Kernel, tiefere Entscheidungsbäume).
  - **Feature Engineering**: Hinzufügen neuer Merkmale oder Transformationen, um die Beziehungen besser abzubilden.
  - **Hyperparameter-Tuning**: Optimierung von Modellparametern, um die Kapazität zu erhöhen.

Ziel

Das Ziel ist ein **ausbalanciertes Modell**, das weder über- noch unterfittet. Ein solches Modell erfasst die zugrunde liegenden Muster der Daten und generalisiert gut auf unbekannte Daten. Hierfür wird oft Cross-Validation eingesetzt, um die Performance zu überwachen und die richtige Komplexität zu wählen.