**3. Please explain the problem of "overfitting" in machine learning, and how you avoid or address it. Please explain from the perspectives of both traditional machine learning and deep learning.**

**convert this content into a table with bullets, remove <br> symbols:**

1. **Overfitting**

Overfitting occurs when a model gets familiar and performs well on training data but fails to generalize to unseen data (i.e., the test/validation set).

In other words, the model “memorizes” the training data rather than learning the underlying features.

Overfitting may bring:

* **High Training Accuracy, Low Test Accuracy:** The model performs exceptionally well on the data it was trained on but struggles when presented with new, unseen data.
* **Large Gap Between Training and Validation Loss:** The model's performance on the training data continues to improve, but its performance on a separate validation dataset starts to worsen.
* **Complex Model:** The model is overly intricate compared to the amount of data available for training.
* **High Variance:** The model's performance fluctuates greatly depending on the specific data it's trained on.
* **Memorization of Noise:** The model is learning irrelevant details or random patterns from the training data instead of the underlying trends.

**(This table is converted to image format to fit the page limits)**

A screenshot of a computer

Description automatically generated

**2a. Overfitting in Traditional Machine Learning**

**Causes:**

* **Complex Models:** Using overly complex models (e.g., high-degree polynomials) can lead to overfitting.
* **Insufficient Data:** When the training dataset is small, the model may fit noise rather than true patterns.
* **Feature Overload:** Including irrelevant or redundant features can exacerbate overfitting.

**Solutions:**

* **Regularization:** Introduce penalties (e.g., L1, L2 regularization) to shrink model coefficients, discouraging extreme values.
* **Cross-Validation:** Split data into training, validation, and test sets. Use the validation set to tune hyperparameters.
* **Feature Selection:** Choose relevant features and discard irrelevant ones.
* **Early Stopping:** Monitor validation loss during training and stop when it starts increasing.
* **Ensemble Methods:** Combine multiple models (e.g., bagging, boosting) to reduce overfitting.

**2b. Overfitting in Deep Learning**

**Causes:**

* **Large Networks:** Deep neural networks with many layers/parameters can easily overfit.
* **Lack of Data/Augmentation:** Insufficient data and augmentation can lead to overfitting.

**Solutions:**

* **Dropout:** Randomly deactivate neurons during training to prevent reliance on specific features.
* **Batch Normalization:** Normalize activations within each mini-batch to stabilize training.
* **Early Stopping:** Monitor validation loss and stop training when it plateaus.
* **Data Augmentation:** Generate additional training samples by applying transformations (e.g., rotation, cropping) to existing data.
* **Transfer Learning:** Use pre-trained models and fine-tune on your specific task.
* **Smaller Networks:** Reduce model complexity if data is limited.