Hyperparameter Tuning

Overfitting occurs when a neural network learns the training data too closely, including its noise, instead of learning general patterns. As a result, the model shows high training accuracy but poor performance on new or unseen data. Since Deep Neural Networks have many parameters, they can easily overfit. Hyperparameter tuning and regularization techniques such as adjusting the learning rate, model complexity, and other settings help reduce overfitting and improve the model's ability to generalize.

Bias-Variance Tradeoff

- **High Bias (Underfitting):** The network is too simple (insufficient capacity) to capture underlying dataset trends.
- **High Variance (Overfitting):** The network model capacity is overly complex, learning random noise, outliers, and spurious correlations in the training set.

Key Strategies for Fixing Overfitting in Neural Networks

To prevent neural networks from overfitting, practitioners apply structural, algorithmic, and data-level regularization techniques:

Weight Regularization (L1 & L2 Regularization)

Weight regularization adds a penalty term proportional to weight magnitude directly to the loss function $L(\theta)$:

- **L1 Regularization (Lasso):** Adds penalty $\lambda \Sigma|w_i|$. Forces non-essential weights strictly to 0, producing sparse feature representations.
- **L2 Regularization (Ridge / Weight Decay):** Adds penalty $\lambda \Sigma(w_i)^2$. Penalizes large weights evenly, encouraging smooth activation functions across all units.

Dropout Regularization

Dropout randomly deactivates a specified fraction p of hidden units during each forward pass step. This prevents co-adaptation among neurons, forcing the network to learn robust redundant features.

Early Stopping

Monitors validation loss during training and halts optimization as soon as validation loss stops improving (or begins rising due to over-memorization), preserving model weights from the optimal epoch.

Data Augmentation & Batch Normalization

- **Data Augmentation:** Artificially expands dataset size and diversity using random transformations (cropping, rotation, flipping, noise injection).

• **Batch Normalization:** Normalizes layer inputs per mini-batch, reducing internal covariate shift and acting as a mild regularizer.

Hyperparameter Search Strategies

Systematic hyperparameter optimization relies on structured search techniques across multi-dimensional search spaces:

1. Grid Search: Exhaustively tests all pre-defined combinations of hyperparameters. Reliable but computationally prohibitive for high dimensions.

2. Random Search: Randomly samples hyperparameter combinations from specified statistical distributions. Highly effective when certain hyperparameters have higher impact than others.

3. Bayesian Optimization (e.g., KerasTuner, Optuna): Constructs a probabilistic surrogate model (Gaussian Process / Tree Parzen Estimator) to predict promising hyperparameter configurations based on past trial results.