TASK 25

Underfitting:

Underfitting occurs when a neural network fails to capture the underlying patterns and relationships in the training data. It typically happens when the model is too simple or lacks the capacity to learn complex patterns. As a result, the network's performance is poor, both on the training data and unseen data.

Signs of underfitting include high training and validation errors, low accuracy, or poor generalization. It suggests that the model is not capturing the true underlying patterns in the data.

To address underfitting in neural networks, you can consider increasing the model's complexity by adding more layers, neurons, or using more advanced architectures. Additionally, increasing the training duration or collecting more training data can also help mitigate underfitting.

Overfitting:

Overfitting occurs when a neural network becomes too specialized in learning the patterns present in the training data to the point that it performs poorly on unseen data. The model becomes overly complex and starts to memorize the training examples, including noise and irrelevant details, rather than generalizing well.

Signs of overfitting include very low training errors but high validation or test errors. The model may show exceptional performance on the training data but fail to generalize to new data.

To combat overfitting, several techniques can be employed, including:

- Regularization: Regularization techniques, such as L1 or L2 regularization, add a penalty term to the loss function to prevent overly large weights in the network. This discourages the model from overemphasizing individual training examples and encourages it to learn more generalizable patterns.

- Dropout: Dropout randomly deactivates a fraction of the neurons during each training iteration, which reduces the model's reliance on specific neurons and encourages robustness and generalization.

- Early stopping: Early stopping involves monitoring the model's performance on a validation set during training and stopping training when the performance starts to deteriorate. This prevents the model from over-optimizing on the training data.

- Data augmentation: Data augmentation techniques, such as rotation, translation, or flipping, can artificially increase the size of the training dataset and introduce variations, reducing the chances of overfitting.

- Model architecture modifications: Simplifying the model architecture, reducing the number of layers or neurons, or applying architectural constraints can help combat overfitting.

Regularization:

Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function during training. The penalty term discourages the model from assigning overly large weights to the parameters, which helps in reducing model complexity and improving generalization.

There are two common types of regularization techniques used in neural networks:

- L1 regularization (Lasso regularization): Adds the sum of absolute values of the weights to the loss function, encouraging the model to use fewer features and produce sparse weights.

- L2 regularization (Ridge regularization): Adds the sum of squared values of the weights to the loss function, penalizing large weight values and encouraging smaller weights across all features.

Regularization helps in controlling the complexity of the model and reducing the impact of irrelevant or noisy features, thus improving generalization performance.

In summary, underfitting occurs when a model is too simple, overfitting happens when a model becomes too complex and specialized to the training data, and regularization techniques are used to control model complexity and prevent overfitting.