Difference between parameter and Hyper-parameter in ML and DL

Parameter vs Hyper-parameter

In machine learning and deep learning, parameters and hyperparameters play distinct roles. Here's the difference between the two:

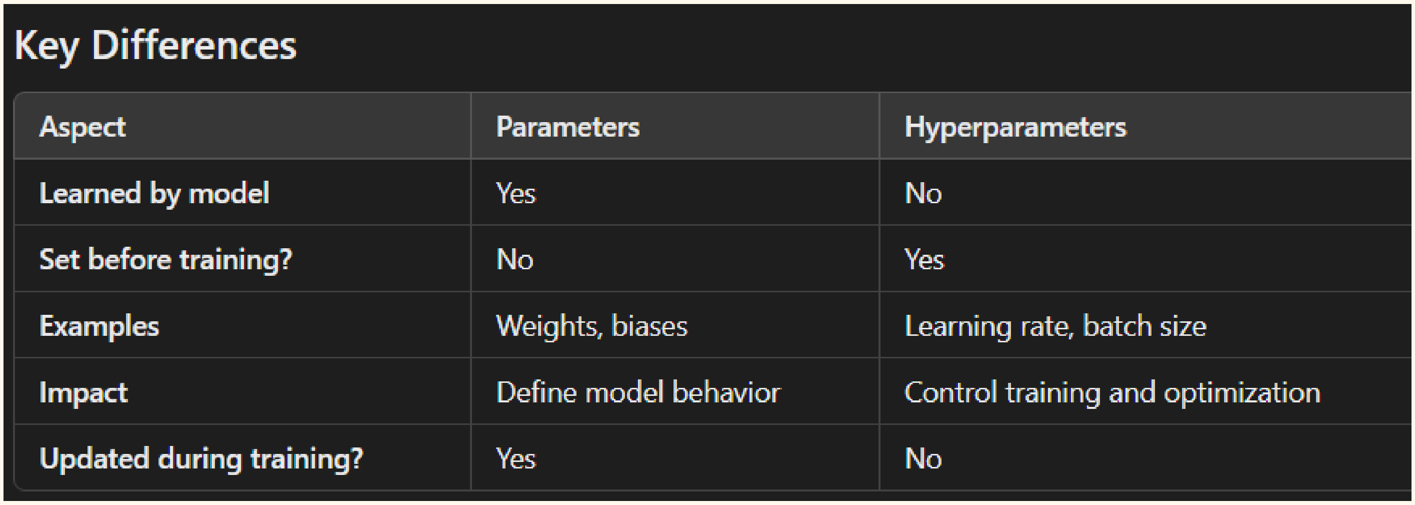
Parameters:

- **Definition**: Parameters are values that the model learns during training. They are adjusted through optimization algorithms like gradient descent to minimize the loss function.
- **Examples**:
  - Weights in a neural network
  - Biases in a neural network
- **Role**: They define the model's internal representation and are directly responsible for making predictions.
- **Updated during training**: Yes, parameters are updated iteratively during the training process.

Hyper-parameters:

- **Definition**: Hyperparameters are settings or configurations defined before the training begins. They are not learned by the model but significantly impact the training process and model performance.
- **Examples**:
  - Learning rate
  - Number of hidden layers or neurons in a neural network
  - Batch size
  - Number of epochs

    - Dropout rate
- **Role**: Hyperparameters control the training dynamics, model capacity, and how the parameters are adjusted.
- **Updated during training**: No, hyperparameters remain fixed during training unless tuned manually or using automated methods like grid search or random search.



Key Differences

Aspect	Parameters	Hyperparameters
Learned by model	Yes	No
Set before training?	No	Yes
Examples	Weights, biases	Learning rate, batch size
Impact	Define model behavior	Control training and optimization
Updated during training?	Yes	No

Role of Hyperparameter Optimization:

Hyperparameters are configuration settings for the training process that are set before training begins and affect how well the model learns. Optimizing these can significantly improve model performance.

Key hyperparameters include:

- Learning rate
- Number of hidden layers and neurons
- Batch size
- Regularization parameters
- Activation functions
- Optimization algorithm

Example of Hyperparameter Optimization:

Example: For an image classification MLP:

- Poor configuration: 1 hidden layer, 10 neurons, learning rate=0.1 → 65% accuracy
- Optimized configuration: 3 hidden layers, 128-64-32 neurons, learning rate=0.001, dropout=0.2 → 85% accuracy

This optimization process might be performed using grid search, random search, or more sophisticated methods like Bayesian optimization to find the best combination of hyperparameters for the specific problem.