

# Introduction to Early Stopping Mechanism

Early stopping is a technique used in machine learning to prevent overfitting by halting the training process once the model's performance stops improving. The key idea is to monitor a performance metric during training, and stop when the metric no longer shows significant improvement over a set number of iterations.

## Key Parameters:

- **Patience (PATIENCE):** This parameter controls how many iterations to wait before stopping if no improvement is observed. If the model doesn't improve for a number of iterations equal to the patience value, training is stopped.
- **Tolerance (TOLERANCE):** This parameter defines the minimum change in the monitored metric to be considered an improvement. If the change is less than this value, the model is considered to have stopped improving.

## Working of Early Stopping:

1. **Initialization:** At the start of training, the patience and tolerance values are set, and a counter is initialized to track how many iterations have passed without improvement.
2. **During Training:** After each iteration, the performance metric is evaluated. If the improvement in the metric is less than the tolerance value, the counter is incremented.
3. **Stopping Condition:** If the counter reaches the patience value, training is stopped early. This prevents the model from overfitting to the training data.

## Advantages of Early Stopping:

- **Prevents Overfitting:** By stopping training before the model begins to overfit, early stopping helps to maintain the model's generalization ability.
- **Saves Resources:** Early stopping can reduce training time and computational costs by preventing unnecessary iterations.
- **Improves Generalization:** By stopping when the model's performance on validation data plateaus, early stopping often results in a model that generalizes better to unseen data.

## Application Example:

In a reinforcement learning scenario, early stopping can be applied by monitoring the change in Q-values and total rewards. If the Q-values and total rewards do not significantly improve over several iterations, training can be stopped to prevent overfitting.

Example Code:

```
const double TOLERANCE = 1e-1;
```

```
const int PATIENCE = 3;
```

```
if (std::fabs(total_reward - previous_total_reward) < TOLERANCE && max_Q_change < TOLERANCE) {
```

```
    no_improvement_count++;
```

```
    if (no_improvement_count >= PATIENCE) {
```

```
        std::cout << "Early stopping at episode " << episode << " due to lack of improvement." << std::endl;
```

```
        break;
```

```
    }
```

```
} else {
```

```
    no_improvement_count = 0;
```

```
}
```