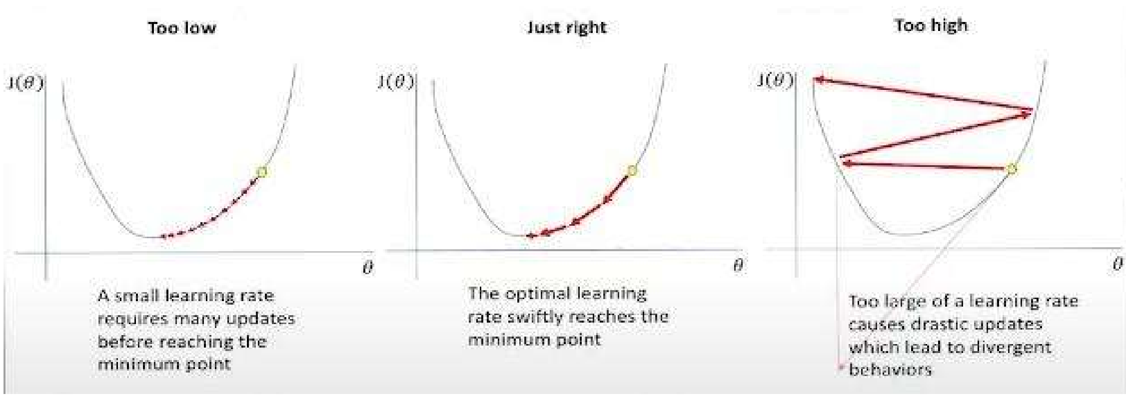
Learning rate (η) is one such hyperparameter that defines the adjustment in the weights of our network concerning the loss gradient descent. It determines how fast or slow we will move towards the optimal weights.

When training a model, the goal is to minimize a loss function, which measures how well the model's predictions match the actual data. Gradient descent achieves this by iteratively adjusting the model's parameters in the direction that reduces the loss function. The learning rate controls how large these adjustments are.

As we know, to update the weights, we use the following function:

$m_{new}=m_{old}-\eta \frac{\partial E}{\partial m}$



Learning Rate Comparison

The Impact of Learning Rate on Training

The value which we choose for learning rate can have the following effects on our training process.

1. Convergence Speed

A well-chosen learning rate accelerates the convergence of the algorithm to the optimal solution. If the learning rate is too high, the algorithm may overshoot the minimum, causing the loss function to diverge or oscillate. Conversely, if the learning rate is too low, the convergence process will be slow, requiring many iterations to reach the optimal solution.

2. Stability and Accuracy

The stability of the training process is directly influenced by the learning rate. A high learning rate can lead to erratic updates, making the training process unstable and the model less accurate. A low learning rate, while stable, might cause the model to get stuck in local minima, potentially leading to suboptimal performance.

3. Generalization

The learning rate also affects the model's ability to generalize to new data. A well-tuned learning rate helps the model find a balance between fitting the training data and maintaining the ability to perform well on unseen data.

Choosing the Right Learning Rate

Selecting an appropriate learning rate is often a matter of experimentation and experience. Here are some common strategies:

1. Grid Search

This involves testing a range of learning rates, often on a logarithmic scale, to identify the best performing value. While this can be computationally expensive, it provides a systematic approach to finding a suitable learning rate.

2. Learning Rate Schedulers

Adaptive learning rate methods adjust the learning rate during training. Techniques like learning rate decay, where the learning rate is reduced over time, or more sophisticated approaches like **ReduceLROnPlateau**, which reduces the learning rate when a metric has stopped improving, can help maintain an optimal learning rate throughout training.

3. Cyclical Learning Rates

This method involves varying the learning rate cyclically between a minimum and maximum value during training. This approach can help the model escape local minima and explore the loss surface more effectively.

4. Adaptive Optimizers

Optimizers such as **Adam**, **RMSprop** and **AdaGrad** adjust the learning rate for each parameter dynamically based on the gradients. These optimizers often require less manual tuning of the learning rate and can lead to faster convergence.