

## task\_8.4

# Machine Learning (WiSe 2025/2026)

Author: Suvansh Shukla

Matriculation No. 256245

---

### Assignment 8 Task 4

The learning rate is a hyper parameter that controls how much to change the model in response to the estimated error each time the model weights are updated. That is, it determines the step size taken in the gradient direction during backpropagation and has direct influence in convergence speed and final accuracy therefore it's considered the most important parameter in this context.

If the learning rate is set too high, the model training can oscillate and become unstable, preventing proper convergence of weights to optimality. If the learning rate is too high, the optimization algorithm (like Gradient Descent) takes steps that are too large. It constantly overshoots the minimum point, causing the loss function's value to jump wildly up and down instead of steadily decreasing. Another scenario termed "Divergence" is also possible, where due to a very high learning rate the steps taken are so large that the algorithm moves away from the minimum, and the loss function value increases indefinitely.

Conversely, a learning rate that is too small significantly increases the latency for model convergence and may result in slow or failed learning. If the learning rate is too low, the steps taken are very small. The algorithm might get stuck in a "valley" (a local minimum) that is not the deepest valley (the global minimum) and take an extremely long time to converge, if at all.

A fixed learning rate may not be a good idea, as in the beginning when the parameters are far from optimal, taking larger steps to reduce the error more quickly. Later on however as the training nears a minimum, smaller steps should be taken to avoid overshooting. Due to such changing circumstances, a single fixed learning rate would not be able to effectively account for both needs. A final catch here would also be that there is no certainty that an optimal learning rate may exist at all.

To cater to the inherent stochastic nature of updates it be more beneficial to customize the learning rate, as per requirement. Taking a larger rate where updates are frequent and taking a smaller rate where updates are more frequent. To do this we may perform Learning Rate Scheduling, where we start out high and then slowly lower the learning rate as the training progresses (thus allowing for fine tuning).

Learning Rate Scheduling has different types:

## Step decay

- Reduces the LR by a factor (e.g., divides by 10) after a fixed number of epochs (e.g., every 30 epochs).
- Simple and effective way to get quick initial progress then settle the loss

## Exponential Decay

- Continuously reduces the LR by an exponential function throughout training.
- Provides a smoother transition from fast learning to fine-tuning.

## Cosine Annealing

- Varies the LR following a cosine curve, starting high and dropping smoothly to a minimum.
- A very popular technique that ensures a very smooth transition.

Another possible approach is to use adaptive optimizers which automatically handle the learning rate adjustment for each parameter individually.