**🎓 Lecture: Adaptive Learning Rate Methods (AdaGrad & RMSprop)**

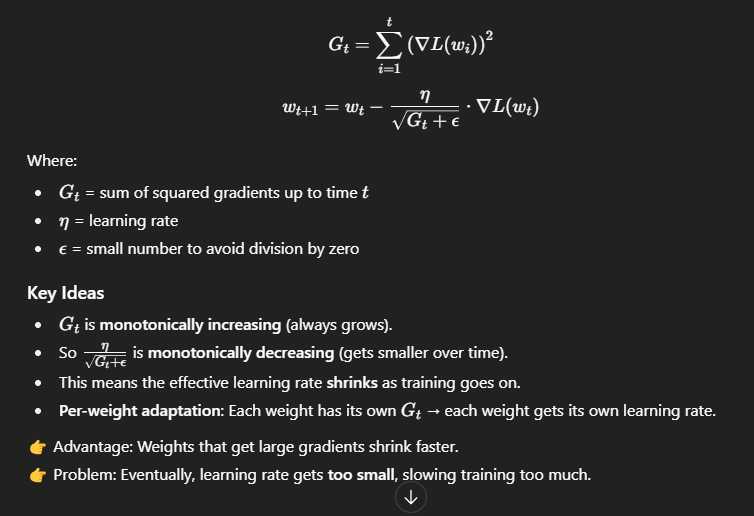
**1. Problem with Fixed Schedules**

* In **Exponential Decay** or **Piecewise Constant schedules**, the learning rate () is reduced following a **predefined rule**.
* But in reality, different weights may need **different step sizes at different times**.
* Solution → **Adaptive methods** adjust the learning rate automatically during training, based on the updates themselves.

**2. AdaGrad (Adaptive Gradient)**

**Formula**

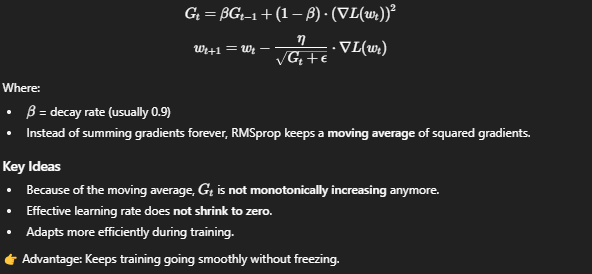
For each parameter :



**3. RMSprop (Root Mean Square Propagation)**

RMSprop is a modification of AdaGrad to fix the problem of the learning rate shrinking too much.

**Formula**



**4. Comparison**

| **Method** | **How Learning Rate Changes** | **Problem / Benefit** |
| --- | --- | --- |
| **AdaGrad** | Always decreases (monotonically) | Can shrink too much → training stalls |
| **RMSprop** | Adapts with moving average (not strictly decreasing) | Keeps learning rate effective throughout training |

**5. Summary**

* **AdaGrad**: First adaptive method, decreases learning rate based on past gradients. Good but may become too small.
* **RMSprop**: Improvement over AdaGrad. Uses moving average with (usually 0.9). Prevents the learning rate from decaying too much.
* Both are **smarter than fixed schedules**, but RMSprop is generally preferred.
* Next step → combine them with momentum → **Adam optimizer** (superior).

✅ **Simple Analogy:**

* **AdaGrad** = You keep track of all your mistakes → eventually, you become overly cautious and stop moving much.
* **RMSprop** = You only remember your **recent mistakes** (moving average), so you adapt but don’t freeze.