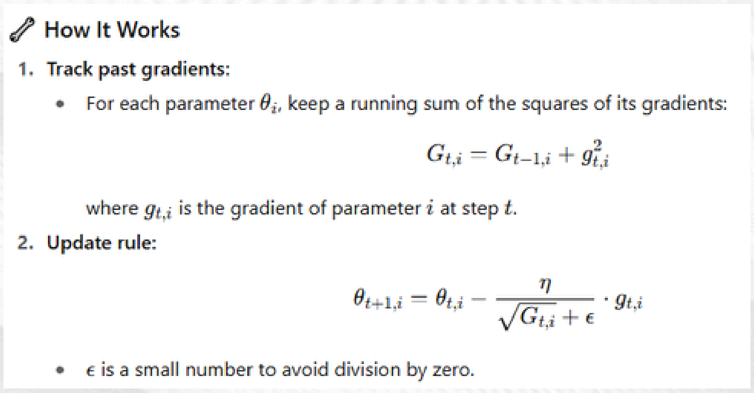
TYPES OF OPTIMIZERS

ADAGRAD

Instead of using a single, fixed learning rate η for all parameters, AdaGrad adapts the learning rate individually for each parameter based on its gradient history.

- If a parameter has large past gradients then reduce its learning rate.
- If a parameter has small past gradients increase its learning rate.
-
- One problem with this is that since it is taking the history of gradients it works fine for first few gradients,
- if the number of gradients increases, the sum of squares for all the gradient gives a very large value.
- Hence this may lead to a learning rate decay problem.



How It Works

1. **Track past gradients:**
   - For each parameter θ_i, keep a running sum of the squares of its gradients:

$$G_{t,i} = G_{t-1,i} + g_{t,i}^2$$

   where $g_{t,i}$ is the gradient of parameter i at step t.

2. **Update rule:**

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i}+\epsilon}} \cdot g_{t,i}$$

   - ϵ is a small number to avoid division by zero.

Analogy:

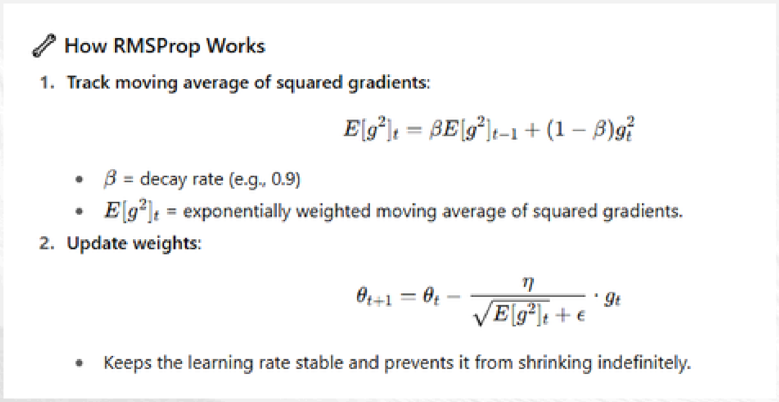
Imagine you're a student learning Math and History.
 You have limited energy (the learning rate) to spend each day.

1. Math (frequent large gradients)
   - You study Math every single day for hours.
   - After a while, you're already very good at it.
   - AdaGrad says: "Stop spending so much time on Math – your improvement will be tiny now."
   - So, it reduces your study time for Math (smaller learning rate).
2. History (rare small gradients)
   - You only study History once in a while.
   - You're still not great at it, so there's a lot to improve.
   - AdaGrad says: "Hey, focus more here – you'll improve faster!"
   - So, it keeps study time high for History (larger learning rate).

RMS PROP

- Instead of using all past gradients, RMSProp uses a moving average of recent squared gradients.
- This means old information fades away, and the learning rate depends only on recent gradient behavior.

- AdaGrad keeps adding up all squared gradients from the start of training.
- Over time, this sum becomes very large learning rate shrinks too much training slows down or stops.



How RMSProp Works

1. **Track moving average of squared gradients:**

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)g_t^2$$

   - β = decay rate (e.g., 0.9)
   - $E[g^2]_t$ = exponentially weighted moving average of squared gradients.

2. **Update weights:**

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t+\epsilon}} \cdot g_t$$

   - Keeps the learning rate stable and prevents it from shrinking indefinitely.

Analogy:

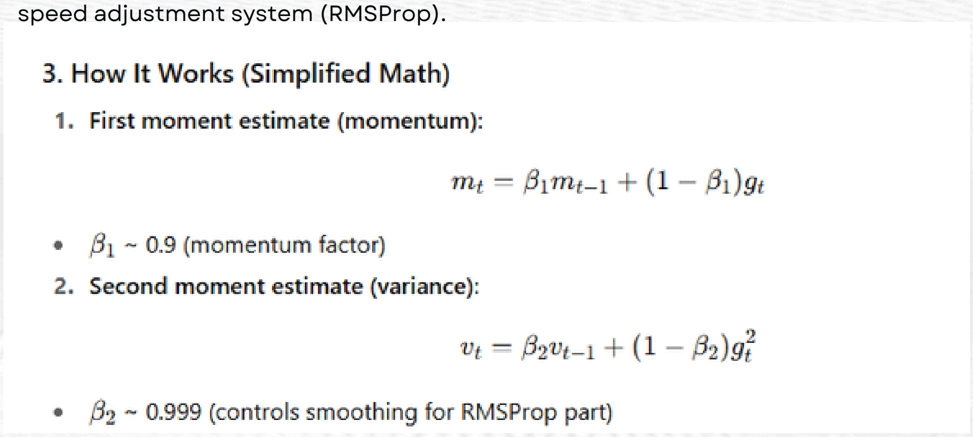
Imagine you're driving:
You adjust your speed based on how bumpy the last few meters were.
If the road has been smooth recently , you speed up.
If it's been bumpy recently , you slow down.
You don't slow down forever just because of bumps 10 miles ago – old bumps fade from memory.

ADAM

Adam was designed to combine the best parts of:

- Momentum → smooths and accelerates learning.
- RMSProp → adapts learning rates for each parameter based on recent gradients.

It's like having both a map of the road ahead (momentum) and a speed adjustment system (RMSProp).



3. How It Works (Simplified Math)

1. **First moment estimate (momentum):**

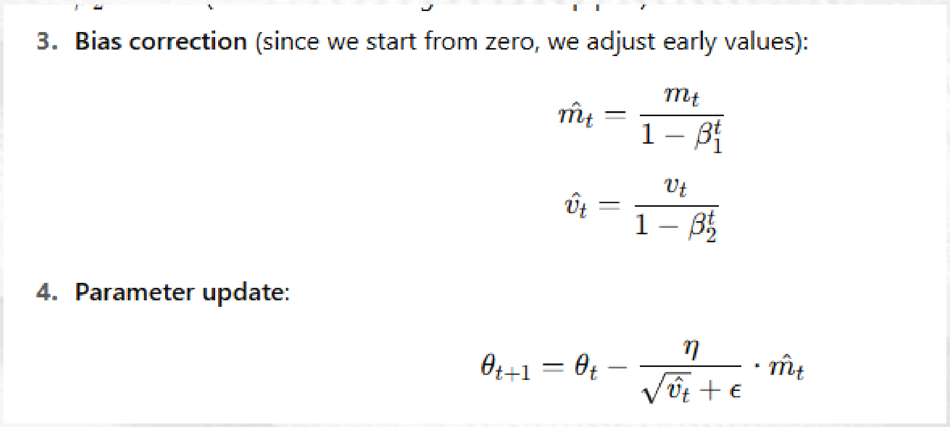
$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

- β_1 ~ 0.9 (momentum factor)

2. **Second moment estimate (variance):**

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

- β_2 ~ 0.999 (controls smoothing for RMSProp part)



3. **Bias correction** (since we start from zero, we adjust early values):

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

4. **Parameter update:**

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t$$

The momentum part helps keep moving in a consistent direction, even if gradients are noisy.
The RMSProp part keeps learning rates adaptive so parameters don't get stuck moving too slow or too fast.
Bias correction ensures early training steps aren't too biased toward zero.

Analogy:

Momentum (GPS memory) → remembers the direction you've been traveling and keeps you going that way.
RMSProp (Speed control) → adjusts your speed based on how bumpy the last part of the road was.
Adam = You have both → You drive smoothly, keep moving in the right direction, and adapt speed smartly depending on road conditions.