ADAM Optimizer

The ADAM (Adaptive Moment Estimation) optimizer combines the ideas of **Momentum** and **RMSProp**. It calculates the exponential moving averages of both the gradients (first moment) and the squared gradients (second moment) and uses them to adaptively adjust the learning rate for each parameter.

Mathematical Equations

1. **Exponential Moving Averages of Gradients and Squared Gradients**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla L(\theta_t)$$

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

- m_t: First moment (mean) of gradients.
- v_t: Second moment (mean of squared gradients).
- β_1 and β_2: Decay rates for m_t and v_t (default values are 0.9 and 0.999).

2. **Bias-Corrected Estimates**

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

3. **Parameter Update Rule**

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

- η: Learning rate.
- ϵ: Small constant for numerical stability (e.g., 10^{-8}).

Example Calculation with Table

Setup

- Loss Function: $L(\theta) = \frac{1}{2}\theta^2$, so $\nabla L(\theta) = \theta$.
- Initial Parameter: $\theta_0 = 2$.
- Hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\eta = 0.1$, $\epsilon = 10^{-8}$.
- Number of Iterations: 5.

Step-by-Step Calculations

Iteration (t)	θ_t	$\nabla L(\theta_t)$	m_t	$\hat{m}_t$	v_t	$\hat{v}_t$	θ_{t+1}
0	2.000	2.000	0.200	2.000	0.004	4.000	1.900
1	1.900	1.900	0.380	2.000	0.0075	4.000	1.805
2	1.805	1.805	0.420	2.05	(..00		

ADAM Optimizer

The ADAM (Adaptive Moment Estimation) optimizer combines **Momentum** and **RMSProp** to create an efficient and adaptive gradient-based optimization algorithm. It maintains exponential moving averages of both the gradients (first moment) and their squares (second moment) and adjusts the learning rate adaptively for each parameter.

Mathematical Equations

1. **Exponential Moving Averages of Gradients and Squared Gradients**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla L(\theta_t)$$

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

- m_t: First moment (mean) of gradients.

- v_t: Second moment (mean of squared gradients).
- β_1 and β_2: Decay rates for m_t and v_t (default values are 0.9 and 0.999).

2. **Bias-Corrected Estimates**

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

3. **Parameter Update Rule**

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

- η: Learning rate.
- ϵ: Small constant for numerical stability (e.g., 10^{-8}).

Example Calculation with Table

Setup

- Loss Function: $L(\theta) = \frac{1}{2}\theta^2$, so $\nabla L(\theta) = \theta$.
- Initial Parameter: $\theta_0 = 2.0$.
- Hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\eta = 0.1$, $\epsilon = 10^{-8}$.

- Number of Iterations: 5.

Step-by-Step Calculation

Iteration (t)	θ_t	$\nabla L(\theta_t)$	m_t	$\hat{m}_t$	v_t	$\hat{v}_t$	θ_{t+1}
1	2.000	2.000	0.200	2.000	0.004	4.000	1.900

Here's the corrected explanation and table for the **ADAM Optimizer** along with its mathematical calculations:

ADAM Optimizer Explanation

The ADAM optimizer calculates an adaptive learning rate for each parameter by combining:

1. **Momentum** (using the first moment, i.e., the moving average of gradients).
2. **RMSProp** (using the second moment, i.e., the moving average of squared gradients).

It includes bias correction to handle the initialization of moving averages.

Mathematical Equations

1. **Exponential Moving Averages of Gradients and Squared Gradients**

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla L(\theta_t)$$

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

- m_t: First moment (mean) of gradients.
- v_t: Second moment (mean of squared gradients).

2. **Bias-Corrected Estimates**

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

3. **Parameter Update Rule**

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

Example Calculation

Setup

- Loss Function: $L(\theta) = \frac{1}{2}\theta^2$, so $\nabla L(\theta) = \theta$.
- Initial Parameter: $\theta_0 = 2.0$.
- Hyperparameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\eta = 0.1$, $\epsilon = 10^{-8}$.
- Iterations: 5.

Step-by-Step Table

Iteration (t)	θ_t	$\nabla L(\theta_t)$	m_t	$\hat{m}_t$	v_t	$\hat{v}_t$	θ_{t+1}
1	2.000	2.000	0.2000	2.0000	0.0040	4.0000	1.9000
2	1.900	1.900	0.3800	2.1000	0.0075	4.2000	1.8100
3	1.810	1.810	0.5410	2.1500	0.0113	4.3500	1.7295
4	1.7295	1.7295	0.6872	2.2000	0.0147	4.4000	1.6568
5	1.6568	1.6568	0.8187	2.2500	0.0179	4.5000	1.5908

Pros and Cons

Pros

1. Combines the strengths of **Momentum** and **RMSProp**, resulting in faster convergence.
2. Handles sparse gradients efficiently.
3. Adaptive learning rate prevents the need for manual tuning.

Cons

1. May not converge to the global minimum but to a good local minimum.
2. Performance is sensitive to hyperparameter settings.
3. The bias correction can introduce additional computation.