Adam Optimizer – Simple Explanation

What is Adam?

Adam stands for **Adaptive Moment Estimation**. It is a smart optimizer used in machine learning and deep learning. Adam improves upon Stochastic Gradient Descent (SGD) by adjusting the learning rate automatically for every parameter during training.

You can think of Adam as **SGD with extra memory and balance:**

- It remembers how the gradients have been changing.
- It adjusts how big or small the next step should be.

Why We Use Adam

1. **Fast and efficient** – learns faster than normal SGD.
2. **Stable** – automatically adjusts step sizes, preventing unstable jumps.
3. **Easy to use** – works well with default parameters.
4. **Ideal for deep learning** – widely used in CNNs, RNNs, and Transformers.

How Adam Works (Step by Step)

Adam keeps track of two values during training:

1. **Average of gradients (momentum):**

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

2. **Average of squared gradients (RMS scaling):**

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

Next, bias correction is applied:

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

Finally, the weights are updated using:

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

Typical constants

$$\beta_1 = 0.9, \quad \beta_2 = 0.999, \quad \epsilon = 10^{-8}, \quad \eta = 0.001$$

Tiny Example

Let us minimize a simple function:

$$L(w) = (w-3)^2$$

so that the gradient is:

$$g = 2(w-3)$$

Step 1

Start with:

$$w_0 = 0, \quad m_0 = 0, \quad v_0 = 0$$

Compute gradient:

$$g_1 = 2(0-3) = -6$$

Compute moving averages:

$$m_1 = 0.9(0) + 0.1(-6) = -0.6$$

$$v_1 = 0.999(0) + 0.001(-6)^2 = 0.036$$

Bias correction:

$$\hat{m}_1 = \frac{-0.6}{1-0.9} = -6, \quad \hat{v}_1 = \frac{0.036}{1-0.999} = 36$$

Weight update:

$$w_1 = 0 - 0.5 \times \frac{-6}{\sqrt{36} + 10^{-8}} = 0.5$$

Now, w has moved closer to the true minimum ($w = 3$).

Comparison with SGD

Feature	SGD	Adam
Learning rate	Fixed	Adapts automatically
Uses past gradients	Optional	Yes
Speed	Slower	Faster
Stability	May oscillate	Stable
Good for	Simple problems	Deep learning

How to Use Adam in Code (Example with PyTorch)

```
import torch
import torch.nn as nn
import torch.optim as optim

# Dummy data: y = 2x
x = torch.tensor([[1.0], [2.0], [3.0], [4.0]])
y = torch.tensor([[2.0], [4.0], [6.0], [8.0]])

# Simple linear model
model = nn.Linear(1, 1)

# Loss function
loss_fn = nn.MSELoss()

# Adam optimizer
optimizer = optim.Adam(model.parameters(), lr=0.1)

# Training loop
for epoch in range(50):
    y_pred = model(x)
    loss = loss_fn(y_pred, y)

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

    if epoch % 10 == 0:
        print(f"Epoch {epoch}, Loss = {loss.item():.4f}")
```

Adam automatically handles learning rate and momentum internally; you only need to specify the optimizer as `optim.Adam(...)`.

Summary

- **What:** Adam is an adaptive optimizer combining Momentum and RMSProp.
- **Why:** It is faster, smoother, and adjusts learning rate automatically.
- **How:** Keeps moving averages of gradients and squared gradients.
- **Where:** Used for deep learning models and large datasets.
- **Example:** Efficiently minimizes functions like $L(w) = (w - 3)^2$.