Optimization Research

Fares Ahmed Moustafa

F.ahmed2270@nu.edu.eg

1. Mathematical Mechanism

1.1. SGD with Momentum

The Stochastic Gradient Descent (SGD) optimizer updates parameters in the direction of the negative gradient of the loss function. However, simple SGD can oscillate, especially in regions where the gradient changes direction rapidly.

Momentum was introduced to solve this problem by adding a "memory" of past gradients. This helps the optimizer move faster in consistent directions and reduces oscillations.

The update rule is:

$$v_t = \beta v_{t-1} + (1-\beta)\nabla_\theta L(\theta_t)$$
$$\theta_{t+1} = \theta_t - \eta v_t$$

This approach smooths the optimization path and helps avoid getting stuck in shallow local minima.

1.2. Adam (Adaptive Moment Estimation)

Adam is one of the most popular adaptive learning rate optimizers in deep learning. It combines the advantages of two earlier algorithms:

- **Momentum** (which keeps track of the first moment, i.e., mean of gradients)
- **RMSProp** (which keeps track of the second moment, i.e., variance of gradients)

Adam uses two moving averages:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_\theta L(\theta_t)$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)(\nabla_\theta L(\theta_t))^2$$

To correct initialization bias (since both $m_t$and $v_t$start at 0), Adam uses:

$$\widehat{m_t} = \frac{m_t}{1-\beta_1^t}, \widehat{v_t} = \frac{v_t}{1-\beta_2^t}$$

Finally, parameters are updated as:

$$\theta_{t+1} = \theta_t - \alpha \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon}$$

Key Difference:
While SGD uses a **global learning rate**, Adam **adapts** the step size **for each parameter** based on its historical gradient behavior.

2. Convergence Analysis

When Adam Converges Faster

Adam often converges faster than SGD, especially during early training or in problems with:

- **Sparse gradients:** common in NLP or high-dimensional image models.
- **Noisy gradients:** such as mini-batch training or irregular loss surfaces.
- **Different parameter scales:** where each parameter benefits from its own adaptive rate.

Adam's automatic adjustment of learning rates helps it navigate complex loss landscapes efficiently, making it ideal for large neural networks like those used in image classification (e.g., CIFAR-10).

When SGD Performs Better

SGD with momentum, although slower initially, tends to reach **flatter minima** in the loss landscape.
Flatter minima are associated with **better generalization**, meaning the model performs better on unseen data.

In contrast, Adam often converges to **sharper minima**, which can lead to slightly worse test accuracy, even if training loss is lower.

3. Generalization & Hyperparameter Tuning

Aspect	SGD with Momentum	Adam
Generalization	Often better test performance once properly tuned	May overfit or generalize slightly worse
Convergence Speed	Slower, may require more epochs	Very fast initial convergence
Learning Rate (η)	Must be tuned carefully; often decayed over time	Automatically adjusted per parameter
Hyperparameters	η (learning rate), β (momentum)	α (learning rate), β□, β□, ε
Ease of Use	Requires scheduling and tuning	Works well “out of the box”
Typical Values	η=0.01, β=0.9	α=0.001, β□=0.9, β□=0.999

In practice:

- **SGD** needs learning rate decay (like step decay or cosine annealing) to reach the best results.
- **Adam** usually works fine with default parameters, which makes it very convenient for experiments.

4. Conclusion and Recommendation

Both optimizers have their strengths and weaknesses.

- **Adam** is best suited for:
    - Quick training or prototyping.
    - Very deep networks.
    - Datasets with sparse or noisy gradients.
    - When tuning time is limited.
- **SGD with Momentum** is preferred for:
    - Final fine-tuning in high-performance image models.
    - Applications requiring maximum generalization accuracy.

- Research or production environments where training stability matters.

Practical Recommendation:
In most industry-level deep learning projects (such as image classification on CIFAR-10 or ImageNet), a **hybrid strategy** is often used:

1. **Start with Adam** for rapid initial convergence.
2. **Switch to SGD with momentum** later in training for improved generalization and stability.

This combines the best of both worlds—fast training and strong final accuracy.