

# Advanced

## Adam Algorithm

- Hạn chế của Gradient Descent: tốc độ học có thể **quá chậm** hoặc **dao động mạnh** nếu chọn sai learning rate

⇒ Giải pháp cải tiến: Adaptive Moment Estimation (Adam)

**Tự động điều chỉnh learning rate** cho mỗi tham số một cách thông minh:

- Nếu tham số đi **đúng hướng liên tục**, tăng  $\alpha_j$  để học nhanh hơn.
- Nếu tham số **dao động qua lại**, giảm  $\alpha_j$  để tránh lặp lại sai lầm.

**Lưu ý khi dùng Adam:**

- Mặc định `learning_rate = 0.001 (1e-3)` thường hoạt động tốt.
- Có thể thử nhiều giá trị khác nhau để đạt hiệu suất học tối ưu hơn.
- Adam **không nhạy cảm** với learning rate như GD, nên dễ dùng hơn.

## Convolutional Neural Network

### Dense Layer (Lớp dày đặc)

- Mỗi neuron trong một layer kết nối với **tất cả các neuron ở layer trước đó**.
- Phù hợp cho nhiều bài toán, nhưng **không tối ưu cho dữ liệu có cấu trúc không gian** (như ảnh hoặc chuỗi thời gian).

### Convolutional Layer (Lớp tích chập)

- Thay vì nhìn toàn bộ input, mỗi neuron chỉ nhìn **một vùng nhỏ (window)** của dữ liệu đầu vào.
- Mỗi neuron học cách **nhận diện các đặc trưng cục bộ** (local features) từ vùng dữ liệu nhỏ.

### Lợi ích:

- **Giảm số lượng tham số** → giúp tính toán nhanh hơn.
- **Ít bị overfitting hơn** → cần ít dữ liệu huấn luyện hơn.

- **Khai thác tính cục bộ và tính dịch chuyển bất biến** trong dữ liệu