

# Di truyền kiến trúc mạng

*Ứng dụng Thuật toán Di truyền tìm kiếm kiến trúc mạng tối ưu*

Nhóm 8 - Neural Architecture Search

Ngày 28 tháng 10 năm 2023

# Mục lục

Mã hóa kiến trúc mạng tích chập

Các toán tử di truyền và giả mã giải thuật

Đánh giá khả năng tìm kiếm

Cải tiến giải thuật

# Kiến trúc mạng truyền thẳng

Một số kiến trúc mạng tích chập hiện đại không tổ chức theo kiểu truyền thẳng (feedforward).  
Với kiến trúc truyền thẳng:

- ▶ Dữ liệu đầu vào của một lớp là dữ liệu đầu ra của lớp liền trước nó.
- ▶ Đa dạng mạng chủ yếu đến từ sự sắp xếp các lớp cơ bản.
- ▶ Muốn tăng thêm độ phức tạp cho mô hình thì bổ sung các lớp khiến cho mạng sâu hơn.

## Hiện tượng thoái biến

Nghiên cứu<sup>1</sup> chỉ ra rằng, việc tăng độ sâu cho mạng không phải lúc nào cũng hiệu quả. Mặc dù không xảy ra quá khớp dữ liệu (overfitting), mạng lại không tốt hơn phiên bản nông của nó. Nguyên nhân là:

- ▶ Hiện tượng bùng nổ, biến mất gradient trong quá trình lan truyền ngược.
- ▶ Không gian hàm số mà mô hình biểu diễn tăng lên, khó khăn trong tìm kiếm tối ưu.

---

<sup>1</sup>Deep Residual Learning for Image Recognition

## Học thặng dư

Xét một mạng có  $L$  lớp. Lớp thứ  $\ell = 1, 2, \dots, L$  tác động lên đầu vào của nó ánh xạ  $\mathcal{H}_\ell(\cdot)$ , là hợp của một số phép toán như tích chập, gộp, kích hoạt ReLU hoặc chuẩn hóa theo lô. Nhiệm vụ của mạng là học ánh xạ này. Đầu ra của lớp thứ  $\ell$  là  $\mathbf{x}_\ell$ .

Với mạng truyền thẳng,  $\mathbf{x}_\ell$  là đầu vào của lớp thứ  $(\ell + 1)$ .

**ResNet**<sup>2</sup> đề xuất, thay vì học trực tiếp ánh xạ  $\mathcal{H}(\mathbf{x})$ , mạng học ánh xạ thặng dư của nó là  $\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x}$  thông qua các kết nối tắt. Giả sử có kết nối tắt từ đầu vào lớp  $\ell$  đến đầu ra lớp  $\ell + k$  (nhảy qua  $k - 1$  lớp), ta học hàm  $\mathcal{F}_{\ell+k}(\cdot)$  mà:

$$\mathbf{x}_{\ell+k} = \mathcal{F}_{\ell+k}(\mathbf{x}_{\ell-1}) + \mathbf{x}_{\ell-1}$$

Trong đó  $\mathcal{F}_{\ell+k} = \mathcal{H}_\ell \circ \mathcal{H}_{\ell+1} \circ \dots \circ \mathcal{H}_{\ell+k}$ , mỗi dãy các hàm như vậy là một khối thặng dư (Residual Block), đơn vị cơ bản của mạng.

---

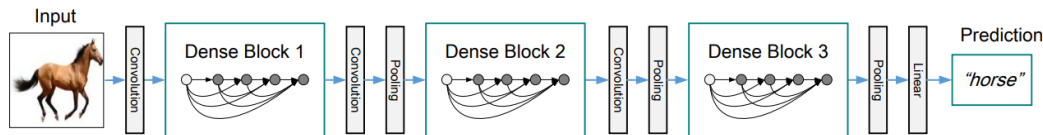
<sup>2</sup>Deep Residual Learning for Image Recognition

## Mạng kết nối dày đặc

**DenseNet**<sup>3</sup> là một kiểu kiến trúc dựa trên kết nối mà bao gồm nhiều Dense Block, trong mỗi Dense Block bắt đầu từ lớp  $\ell$  và kết thúc bởi lớp  $\ell + K$  thì với mọi  $k = 1, 2, \dots, K$ , ta học ánh xạ  $\mathcal{H}_{\ell+k}(\cdot)$  mà:

$$\mathbf{x}_{\ell+k} = \mathcal{H}([\mathbf{x}_{\ell}, \mathbf{x}_{\ell+1}, \dots, \mathbf{x}_{\ell+k-1}])$$

Nghĩa là đầu ra của một lớp trong khối dựa trên đầu ra của tất cả lớp trước nó trong khối. Điều này được hiểu như đang tái sử dụng đặc trưng đã trích xuất được từ các lớp trước nó cho việc học.



Hình 1: Minh họa kiến trúc DenseNet

<sup>3</sup>Densely Connected Convolutional Networks

## Khảo sát kiến trúc mạng

Xét mạng với số lượng lớp hạn chế, ta sẽ mô tả không gian các kiến trúc có thể có của nó.

- ▶ Mạng bao gồm tất cả  $S$  giai đoạn.
- ▶ Giai đoạn thứ  $s = 1, 2, \dots, S$  có  $K_s$  nút là  $v_{s,k_s}$  với  $k_s = 1, 2, \dots, K_s$ .
- ▶ Mỗi nút là liên tiếp các phép toán: tích chập, chuẩn hóa theo lô<sup>4</sup> và kích hoạt ReLU.
- ▶ Liên sau mỗi giai đoạn là lớp gộp và kết thúc mạng là lớp kết nối đầy đủ.

Trong mỗi giai đoạn thì các nút có kết nối với nhau, tức là đầu ra của nút đứng trước có thể được sử dụng cho đầu vào của các nút đứng sau. So với ResNet, kết nối chỉ là dạng cộng trực tiếp, hay DenseNet có kết nối tất cả giữa các nút trước đến sau.

---

<sup>4</sup>Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift ▶ ◀ ≡ ▶ ≡ ≡ ◀ ▶ ◀ ▶

## Tổ chức kiến trúc mạng

Trong giai đoạn  $s$ , ta sử dụng  $\frac{1}{2}K_s(K_s - 1)$  bit nhị phân để biểu diễn.

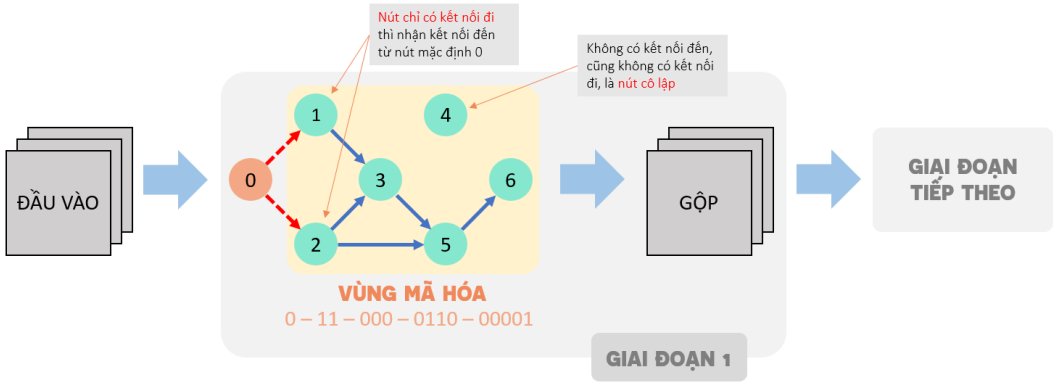
- ▶ 1 bit đầu tiên thể hiện kết nối  $v_{s,1}$  đến  $v_{s,2}$ .
- ▶ 2 bit tiếp theo thể hiện kết nối  $v_{s,1}, v_{s,2}$  đến  $v_{s,3}$ .
- ▶ ...
- ▶  $K_s - 1$  bit cuối cùng thể hiện kết nối  $v_{s,1}, v_{s,2}, \dots, v_{s,K_s-1}$  đến  $v_{s,K_s}$ .

Để đảm bảo kiến trúc hợp lệ, ta bổ sung nút thứ 0 bắt đầu mỗi giai đoạn:

- ▶ Nút 0 vẫn có cấu tạo gồm: tích chập, chuẩn hóa theo lô, ReLU.
- ▶ Đầu ra của nút 0 chuyển đến những nút mà chỉ có kết nối đi, chưa có kết nối đến.
- ▶ Nếu mã hóa là một dãy chỉ số 0, thì giai đoạn vẫn có ít nhất một nút là nút 0.



# Triển khai mã hóa



Hình 2: Minh họa giai đoạn trong kiến trúc mạng và mã hóa

## Nguyên tắc khởi tạo

Quá trình di truyền bắt đầu với một quần thể gồm  $N$  cá thể ngẫu nhiên. Sau đó, thực hiện  $T$  vòng lặp (hay  $T$  thế hệ), mỗi thế hệ gồm 3 hoạt động: chọn lọc, lai ghép, đột biến. Hàm thích nghi của mỗi cá thể được đánh giá thông qua việc đào tạo từ đầu trên tập dữ liệu tham chiếu.

- ▶ Khởi tạo quần thể các mô hình  $\{\mathbb{M}_{0,n} : n = 1, 2, \dots, N\}$
- ▶ Mỗi mô hình là một chuỗi nhị phân  $L$  bit, tức:

$$\mathbb{M}_{0,n} : \mathbf{b}_{0,n} = [b_{0,n}^1, b_{0,n}^2, \dots, b_{0,n}^L] \in \{0, 1\}^L$$

- ▶ Mỗi bit trong cá thể được lấy mẫu độc lập từ một phân phối Bernoulli:

$$\ell = 1, 2, \dots, L : b_{0,n}^\ell \sim \mathcal{B}(L, 0.5)$$

Các chiến lược khởi tạo khác nhau không ảnh hưởng nhiều đến hiệu suất di truyền. Ngay cả khi khởi tạo ngẫu nhiên (tất cả các cá thể đều là chuỗi 0), quá trình di truyền có thể khám phá ra các kiến trúc cạnh tranh nếu số thế hệ là đủ lớn.

## Nguyên tắc chọn lọc

Quá trình chọn lọc được thực hiện ở đầu mỗi thế hệ.

- ▶ Trước thế hệ thứ  $t$ , cá thể thứ  $n$  là  $\mathbb{M}_{t-1,n}$  được gán một hàm thích nghi.
- ▶ Hàm thích nghi được định nghĩa bởi tỷ lệ nhận dạng  $r_{t-1,n}$  và tác động trực tiếp lên khả năng  $\mathbb{M}_{t-1,n}$  sống sót qua quá trình chọn lọc.

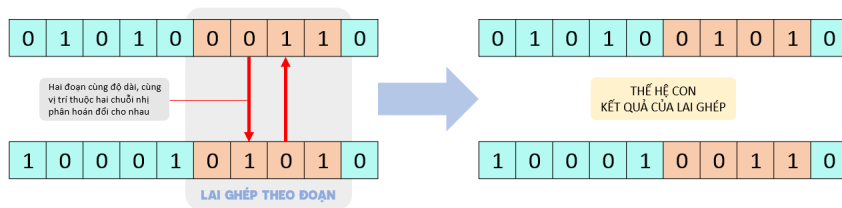
Cuộc chiến sinh tồn diễn ra sau đó.

- ▶ Mỗi cá thể trong thế hệ tiếp theo  $\mathbb{M}_{t,n}$  được xác định một cách độc lập thông qua một lấy mẫu không đồng đều trên tập  $\{\mathbb{M}_{t-1,n} : n = 1, 2, \dots, N\}$ .
- ▶ Xác suất lấy mẫu  $\mathbb{M}_{t-1,n}$  tỷ lệ thuận với  $(r_{t-1,n} - r_{t-1,0})$ , trong đó  $r_{t-1,0}$  là độ chính xác nhận dạng nhỏ nhất trong tất cả thế hệ trước đó.

Có nghĩa, cá thể tốt nhất có xác suất lớn nhất để được chọn, và cá thể tồi tệ bị loại bỏ. Vì số lượng cá thể  $N$  không thay đổi, mỗi cá thể trong thế hệ trước có thể được chọn nhiều lần.

## Toán tử lai ghép

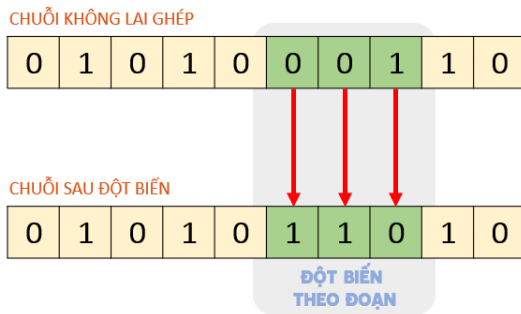
- ▶ Quá trình lai ghép thay đổi 2 cá thể cùng lúc.
- ▶ Thay vì xem xét từng bit một cách riêng lẻ, đơn vị cơ bản trong lai ghép là một đoạn (một số bit liên tiếp nhau). Tham số sử dụng là bán kính lai ghép  $r_C$ .
- ▶ Mỗi cặp đoạn được trao đổi với xác suất nhỏ  $p_C$ .



Hình 3: Minh họa phép lai ghép 2 điểm cắt với bán kính lai ghép 4

## Toán tử đột biến

- ▶ Với những cá thể trải qua lai ghép thì thực hiện đột biến.
- ▶ Quá trình này đảo bit của chuỗi với một xác suất nhỏ  $p_M$  và tác động lên một đoạn có độ dài phụ thuộc bán kính đột biến  $r_M$ .



Hình 4: Minh họa phép đột biến với bán kính đột biến 3

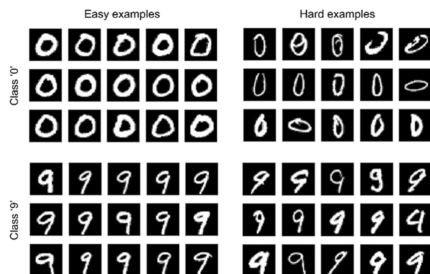
## Tóm tắt thuật toán

1. **Đầu vào:** Tập dữ liệu tham chiếu  $\mathcal{D}$ , số lượng thế hệ  $T$ , số lượng cá thể trong mỗi thế hệ  $N$ , xác suất đột biến và lai ghép là  $p_M$  và  $p_C$ , bán kính đột biến và lai ghép là  $r_M$  và  $r_C$ .
2. **Khởi tạo:** Bắt đầu với một tập các cá thể ngẫu nhiên  $\{\mathbb{M}_{0,n} : n = 1, 2, \dots, N\}$ , và tính toán độ chính xác nhận diện của từng cá thể.
3. Trong mỗi vòng lặp  $t = 1, 2, \dots, T$  thực hiện các thao tác:
  - ▶ **Chọn lọc:** Chọn cha mẹ tham gia lai ghép theo thể thức *Roulette Wheel*.
  - ▶ **Lai ghép:** Với mỗi cặp cha mẹ chọn được, thực hiện lai ghép với xác suất  $p_C$  và tham số  $r_C$ .
  - ▶ **Đột biến:** Với mỗi cá thể sau lai ghép, thực hiện đột biến với xác suất  $p_M$  và tham số  $r_M$ .
  - ▶ **Đánh giá:** Tính toán độ chính xác nhận dạng cho từng cá thể mới.
  - ▶ **Đấu tranh:** Bỏ sung thế hệ con vào quần thể và chọn ra những cá thể thích nghi tốt nhất cho thế hệ tiếp theo.
4. **Đầu ra:** Một tập hợp các cá thể trong thế hệ cuối cùng  $\{\mathbb{M}_{T,n} : n = 1, 2, \dots, N\}$  cùng độ chính xác nhận dạng của chúng trên  $\mathcal{D}$ .

## Chữ số viết tay: MNIST

- ▶ Là ảnh xám, chỉ có một kênh màu duy nhất, kích thước  $28 \times 28 \times 1$ .
- ▶ Có 10 lớp phân loại, là các chữ số 0 đến 9.

Trong khoảng 100 ảnh cuối cùng của bộ MNIST là rất khó để nhận dạng chính xác, yêu cầu kiến trúc rất sâu và hết sức phức tạp. Tỷ lệ nhận dạng chính xác trên tập kiểm thử tốt nhất hiện nay vào khoảng 99.7% đến 99.84%.



### Hình 5: Những trường hợp nhận dạng chính xác và khó khăn

## Thiết kế thuật toán

Thuật toán Di truyền yêu cầu tài nguyên tính toán tương đối lớn. Để đơn giản, ta chỉ xem xét trường hợp hạn chế của nó:

- ▶ Có tất cả 10 thể hệ, mỗi thể hệ có 7 cá thể.
- ▶ Chọn tham số  $r_C = 4, r_M = 2$  để bảo toàn kiến trúc địa phương của mạng và không thay đổi mạng quá nhiều.
- ▶ Các xác suất  $p_C = 0.7, p_M = 0.2$  để sinh ra đa dạng kiến trúc cho quần thể.

Kiến trúc mạng được thiết kế:

- ▶ Chỉ gồm đúng 1 giai đoạn, sau đó là gộp và kết nối đầy đủ.
- ▶ Có tất cả 7 nút trong mỗi giai đoạn, mã hóa bởi chuỗi nhị phân dài 21.
- ▶ Chỉ cập nhật tham số cho mạng bằng 5 vòng lặp với trình tối ưu ADAM<sup>5</sup>, tốc độ học mặc định 0.01.

---

<sup>5</sup>Adam: A Method for Stochastic Optimization



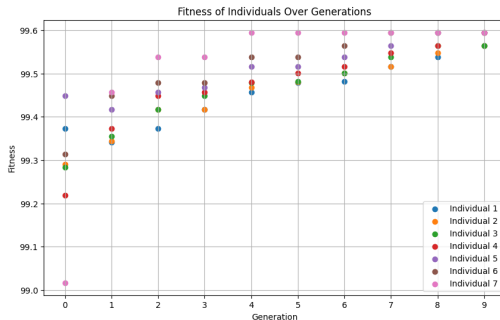
## Tìm kiếm kiến trúc MNIST

Nhận xét về kết quả thu được:

- Quần thể hội tụ sau khoảng 10 thế hệ.
- Ở những cá thể lân cận tối ưu, sự thay đổi độ thích nghi sau mỗi thế hệ không còn đáng kể.

Hạn chế của thuật toán:

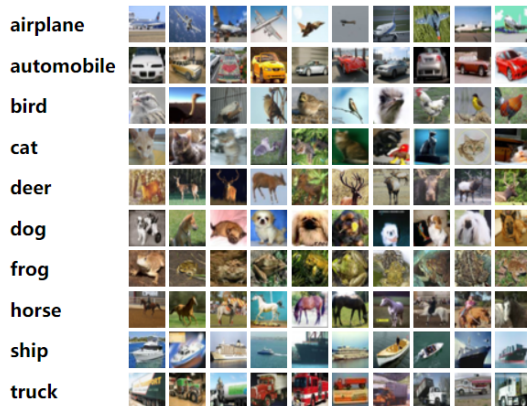
- Không khẳng định được kết quả cuối cùng là tối ưu toàn cục.
- Đa dạng trong quần thể bị giảm đi do áp dụng chọn lọc theo thứ hạng, một cá thể có khả năng xuất hiện nhiều lần gây ảnh hưởng đến quá trình di truyền.



Hình 6: Độ thích nghi của cá thể qua các thế hệ

## Phân loại ảnh màu: CIFAR-10

- ▶ Là ảnh màu, có 3 kênh màu RGB, kích thước  $32 \times 32 \times 3$ .
- ▶ Có 10 lớp phân loại, hình ảnh động vật hoặc thực thể.



Hình 7: Một số mẫu trên bộ dữ liệu CIFAR-10

# Thiết kế thuật toán

Trường hợp được xét hạn chế với các điều kiện:

- ▶ Có tất cả 10 thể hệ, mỗi thể hệ có 7 cá thể.
- ▶  $r_C = 4, r_M = 2, p_C = 0.7, p_M = 0.2$  được chọn.

Kiến trúc mạng được thiết kế:

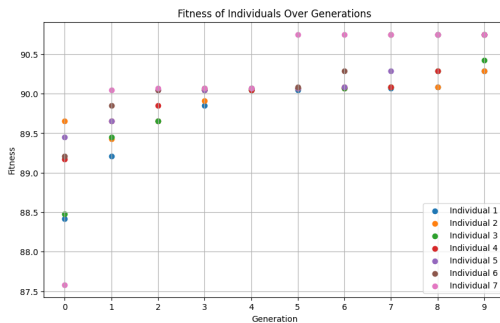
- ▶ Chỉ gồm 2 giai đoạn, kết thúc mỗi giai đoạn là gộp, kết thúc mạng là kết nối đầy đủ.
- ▶ Có 5 nút trong mỗi giai đoạn, mã hóa toàn mạng bởi chuỗi nhị phân dài 20.
- ▶ Cập nhật tham số cho mạng bằng 5 vòng lặp.

## Tìm kiếm kiến trúc CIFAR-10

Nhận xét về kết quả thu được:

- ▶ Có một giai đoạn trong đó quần thể thoát khỏi tối ưu địa phương và đạt đến tối ưu tốt hơn nhiều,
- ▶ Quần thể hội tụ sau khoảng 10 thế hệ khi có một cá thể chiếm đại đa số và mức độ thay đổi không nhiều.

Thuật toán đạt được hiệu quả nhất định, nhưng chưa khắc phục được vấn đề một cá thể bao vây quần thể, làm giảm không gian tìm kiếm và mất mát đa dạng nhiễm sắc thể.



Hình 8: Độ thích nghi của cá thể qua các thế hệ

# Khắc phục các hạn chế

## 1. Chỉ duy trì một tập con các cá thể tốt nhất qua các thế hệ.

- ▶ **Di truyền ưu tú (Elitism):** Chỉ một hay một vài cá thể tốt nhất được chọn lại, không chọn theo thứ hạng. Tăng đa dạng nhưng vẫn đảm bảo thế hệ sau không tệ hơn thế hệ trước.
- ▶ **Chia sẻ thích nghi (Fitness Sharing):** Tính toán khoảng cách Hamming giữa các cá thể, trừng phạt cá thể bằng cách giảm độ thích nghi dựa trên số cá thể tương đồng với nó. Đảm bảo một cá thể và các phiên bản tương tự nó không xuất hiện quá nhiều.

## 2. Tăng đa dạng nhiễm sắc thể trong quần thể.

- ▶ **Quần thể động (Dynamic Population Size):** Tăng dần kích thước quần thể theo thời gian, để đủ sức chứa cả những cá thể khác.
- ▶ **Đa quần thể (Multiple Populations):** Sử dụng nhiều quần thể riêng biệt và cho chúng tương tác nhau theo một lịch trình.
- ▶ **Tai họa đột biến (Cataclysmic Mutation):** Khi thuật toán mắc kẹt, áp dụng một tỷ lệ đột biến cao cho toàn bộ quần thể.

# Tóm tắt thuật toán

1. Khởi tạo 3 quần thể cùng kích thước theo phân phối Beurnoulli  $\mathcal{B}(0.25), \mathcal{B}(0.5), \mathcal{B}(0.75)$ .
2. Di truyền ưu tú quần thể  $\mathcal{B}(0.5)$ :
  - 2.1 Truyền trực tiếp 1 cá thể tốt nhất qua thế hệ tiếp theo.
  - 2.2 Các cá thể còn lại lai ghép, đột biến tạo ra cá thể con.
  - 2.3 Bổ sung cá thể con vào quần thể và tính toán hàm thích nghi (đã chia sẻ bớt).
  - 2.4 Chọn lọc dựa trên hàm thích nghi và hình thành thế hệ mới.
  - 2.5 Sau một số thế hệ, nếu quần thể không đa dạng thì tiến hành di trú cá thể từ quần thể khác đến và tăng kích thước tối đa.
3. Kết thúc nếu quần thể đã tăng đến kích thước lớn và tìm ra những cá thể tốt nhất.