# Chương 3

# LÝ THUYẾT NỀN TẢNG

Ứng dụng chụp ảnh giải trí sẽ biến đổi hình ảnh một khuôn mặt người sang các hình thái khác nhau: từ trẻ thành già, từ nam thành nữ, từ không có râu thành có râu ... Những phép biến đổi trên rất là khó và mất nhiều thời gian nếu sử dụng các cách chỉnh sửa ảnh thủ công truyền thống. Vì vậy, ứng dụng chụp ảnh giải trí sẽ sử dụng áp dụng trí tuệ nhân tạo để biến đổi hình ảnh khuôn mặt trên. Cụ thể mô hình được áp dụng ở đây sẽ là StarGan, một dạng biến thể từ mô hình đối nghịch tạo sinh.

## 3.1 Lý thuyết nền tảng mô hình đối nghịch tạo sinh

### 3.1.1 Tổng quan

Mô hình đối nghịch tạo sinh là một mô hình mạng nơ-ron nhân tạo sử dụng các tích chập để huấn luyện trên tập dữ liệu hình ảnh nhưng không giống như tích chập thông thường, mô hình đối nghịch tạo sinh hoạt động theo cách thức chơi một trò chơi min max giữa hai mạng: mạng tạo và mạng phân biệt.

### 3.1.2 Kiến thức nền tảng về mạng nơ-ron

- Mạng nơ-ron đa lớp được tạo thành bởi nhiều nơ-ron được tổ chức thành các lớp. Mạng nơ-ron đa lớp thường có 3 thành phần: lớp đầu vào ( Input layer), lớp ẩn (Hidden layer), lớp đầu ra (Output layer). Mạng nơ-ron đa lớp thường được gọi là mạng nơ-ron truyền thẳng (feedforward neural network). Mục tiêu của mạng nơ-ron đa lớp là ước lượng các hàm $f^*$ . Ví dụ, đối với bài toán phân lớp, $y = f^*(x)$ gán đầu vào x đến nhóm y. Một mạng nơ-ron đa lớp sẽ định nghĩa một hàm $y = f(x;\theta)$ và học giá trị của tham số $\theta$ để được kết quả là một hàm tương đương tốt nhất. Mạng nơ-ron đa lớp đóng vai trò quan trọng trong máy học, nó có mặt trong nhiều ứng dụng thương mại. Điển hình như mạng tích chập (convolutional networks) dùng để nhận dạng vật thể trong hình ảnh là một dạng đặc biệt của mạng nơ-ron đa lớp.

- Lan truyền ngược: Quá trình sử dụng mạng nơ-ron truyền thẳng để nhận một đầu vào x và sinh ra một đầu ra $\hat{y}$ được gọi là lan truyền thẳng (forward propagation). Trong quá trình huấn luyện, lan truyền thẳng có thể tiếp tục cho tới khi nó tạo ra một chi phí $J(\theta)$. Thuật toán lan truyền ngược (back-propagation) cho phép thông tin từ chi phí chảy ngược lại trong mạng, để tính toán độ dốc (gradient).

- Không gian ẩn (Latent space) là một không gian mà các điểm dữ liệu tương đồng nhau thì sẽ ở gần nhau hơn.

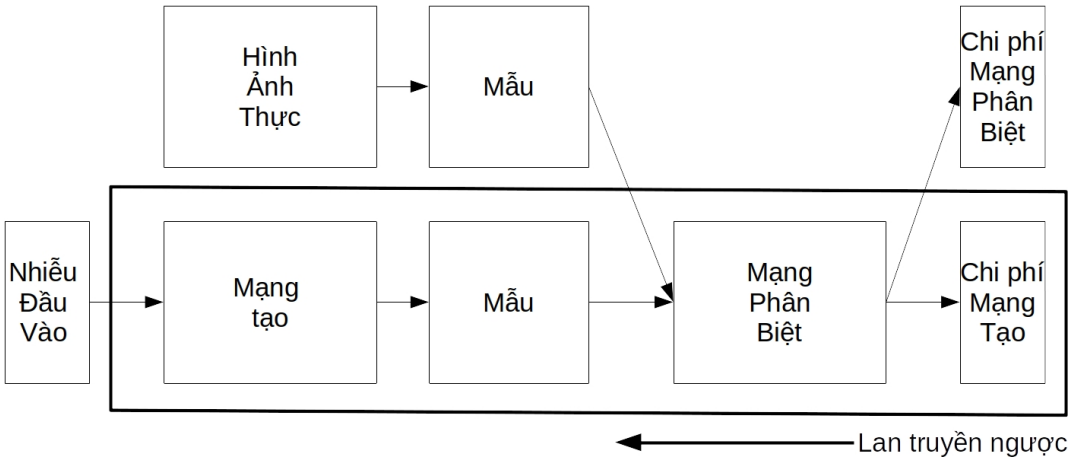
- Mã ẩn (Latent code):là một vectơ đầu vào chứa các dữ liệu nén.

### 3.1.3 Mạng đối nghịch tạo sinh GAN

- Mạng tạo (Generator): Thành phần mạng tạo của GAN học cách tạo dữ liệu đầu ra bằng cách kết hợp phản hồi từ mạng phân biệt. Mạng tạo học cách làm cho mạng phân biệt trả về kết quả là thực

đối với dữ liệu mà mạng tạo đã tạo ra. Quá trình đào tạo mạng tạo yêu cầu tích hợp chặt chẽ giữa mạng tạo và mạng phân biệt. Quy trình đào tạo mạng tạo như sau:

- Lấy mẫu nhiễu ngẫu nhiên.
- Mạng tạo biến đầu vào ngẫu nhiên thành một phiên bản dữ liệu mới.
- Mạng phân biệt sẽ suy xét phiên bản dữ liệu trên là thực hay giả.
- Tính toán chi phí từ mạng phân biệt.
- Lan truyền ngược thông qua cả mạng phân biệt và mạng tạo để thu được các gradient.
- Sử dụng gradient để thay đổi các trọng lượng của mạng tạo.



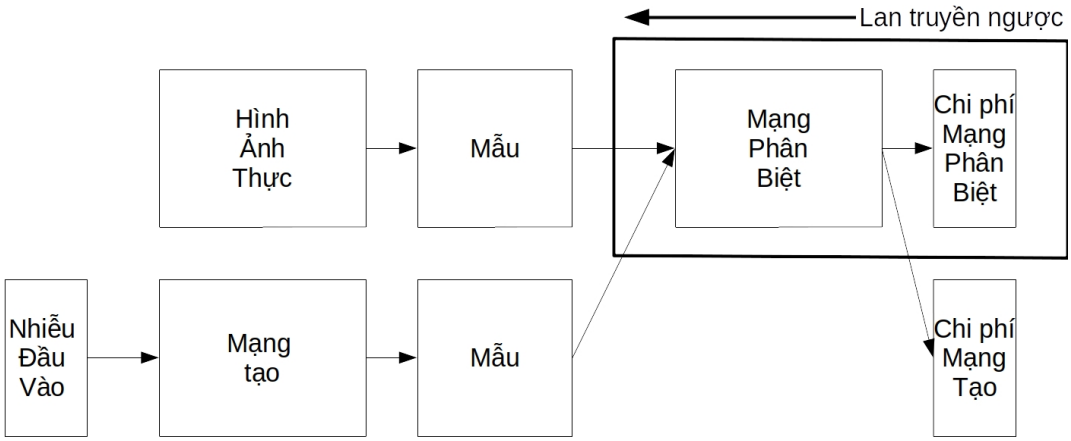

Hình 3.1: Sự lan truyền ngược trong đào tạo mạng tạo

Để đào tạo một mạng nơ-ron, chúng ta thay đổi trọng số của mạng để giảm lỗi hoặc chi phí đầu ra của nó. Tuy nhiên, trong GAN, mạng tạo không được kết nối trực tiếp với chi phí mà chúng ta đang cố gắng tác động, mà chi phí trên sẽ được tính toán và tác động bởi mạng phân biệt. Lan truyền ngược điều chỉnh từng trọng lượng theo

đúng hướng bằng cách tính toán tác động của trọng lượng trên đầu ra.

- Đầu vào ngẫu nhiên : Ở dạng cơ bản nhất, GAN lấy nhiễu ngẫu nhiên làm đầu vào của nó. Sau đó mạng tạo biến nhiễu này thành một đầu ra có ý nghĩa. Bằng cách đưa vào nhiều loại nhiễu khác nhau, GAN tạo ra nhiều loại dữ liệu. Các thực nghiệm cho thấy dạng phân phối của nhiễu không quan trọng lắm, vì vậy chúng ta có thể chọn thứ gì đó dễ lấy mẫu, chẳng hạn như phân phối đồng đều. Để thuận tiện, không gian mà nơi nhiễu được lấy mẫu thường có kích thước nhỏ hơn kích thước của không gian dầu ra.

- Mạng phân biệt (Discriminator): Mạng phân biệt trong mô hình GAN đơn giản chỉ là một bộ phân lớp. Mạng phân biệt sẽ suy xét xem dữ liệu trả về từ mạng sinh là thật hay giả. Mạng phân biệt có thể sử dụng bất kỳ kiến trúc mạng phù hợp với dạng dữ liệu mà nó phân lớp.




Hình 3.2: Sự lan truyền ngược trong đào tạo mạng phân biệt

Dữ liệu huấn luyện được lấy từ các nguồn:

  - Dữ liệu thực: ví dụ như ảnh của một người.
  - Dữ liệu giả: phiên bản dữ liệu được tạo ra bởi mạng tạo.

Trong hình 2.2, mạng phân biệt kết nối đến hai hàm chi phí. Trong quá trình huấn luyện, mạng phân biệt bỏ qua chi phí mạng tạo và chỉ sử dụng chi phí mạng phân biệt. Chí phí mạng tạo sẽ được sử dụng trong quá trình đào tạo mạng tạo. Quá trình đào mạng phân biệt được mô tả như sau:

- Mạng phân biệt phân lớp dữ liệu thật và dữ liệu giả được tạo ra bởi mạng tạo.
- Mạng phân biệt dựa vào hàm chi phí để cập nhật các trọng số thông bằng quá trình lan truyền ngược.

### 3.1.4 Mạng nơ-ron tích chập CNN

- Filter layer: Lớp lọc sẽ lọc đầu vào để tạo nên bản đồ các đặc trưng, bản đồ này tóm tắt sự hiện diện của các đặc trưng đã được phát hiện trong đầu vào.
- Pool layer: Lớp gộp là một lớp mới được thêm vào sau lớp tích chập. Các lớp trong một mô hình có thể trông như sau:
  - Ảnh đầu vào
  - Lớp tích chập
  - Phi tuyến tính
  - Lớp gộp

  Việc bổ sung lớp gộp sau lớp tích chập là mẫu phổ biến được sử dụng để sắp xếp thứ tự các lớp trong mạng nơ-ron có thể được lặp lại một hoặc nhiều lần trong một mô hình nhất định. Lớp gộp hoạt động dựa trên từng bản đồ đặc trưng riêng biệt để tạo một tập hợp mới với cùng số lượng bản đồ đặc trưng được gộp chung.
  - Avg Pool: Tính giá trị trung bình cho mỗi bản vá trên bản đồ đặc trưng.

  - Max Pool: Tính giá trị lớn nhất cho mỗi bản vá của bản đồ đặc trưng.

- Residual Block (Skip connection): Các khối dư là các khối bỏ qua kết nối, chúng học các hàm còn lại với tham chiếu đến các đầu vào của lớp thay vì học các hàm không tham chiếu. Chúng được giới thiệu như một phần của kiến trúc ResNet. Về mặt hình thức, biểu thị ánh xạ cơ bản mong muốn là H(x), các lớp phi tuyến xếp chồng lên nhau phù hợp với một ánh xạ khác của $F(x) := H(x) - x$. Ánh xạ ban đầu được biểu diễn lại thành F(x)+x. x hoạt động như phần dư, vì vậy được gọi là khối dư. Có thể hình dung việc tối ưu hóa ánh xạ phần dư sẽ dễ dàng hơn là tối ưu hóa ánh xạ gốc, không tham chiếu. Ở mức cực đoan, nếu một ánh xạ nhận dạng là tối ưu, sẽ dễ dàng đẩy phần dư về 0 hơn là phù hợp với một ánh xạ nhận dạng bởi một chồng các lớp phi tuyến. Việc bỏ qua các kết nối cho phép mạng dễ dàng tìm hiểu các ánh xạ giống như danh tính.

- Fully connected layer: Mục tiêu của lớp được kết nối đầy đủ là lấy kết quả của quá trình tích chập / gộp và sử dụng chúng để phân loại hình ảnh thành một nhãn. Đầu ra của tích chập / gộp được làm phẳng thành một vectơ giá trị duy nhất, mỗi vectơ đại diện cho xác suất một đối tượng nào đó thuộc về một nhãn. Ví dụ: nếu hình ảnh là một con mèo, các đặc điểm đại diện cho những thứ như râu hoặc lông sẽ có xác suất cao đối với nhãn "mèo".

### 3.1.5 Hàm kích hoạt Activation function

Trong một mạng nơ-ron, hàm kích hoạt chịu trách nhiệm chuyển tổng trọng số đầu vào thành tín hiệu kích hoạt của nút hoặc đầu ra cho đầu vào đó.

- ReLU: Trong mạng nơ-ron, hàm kích hoạt chịu trách nhiệm biến đổi tổng trọng số đầu vào từ nút thành sự kích hoạt của nút hoặc đầu

ra cho đầu vào đó. Hàm kích hoạt tuyến tính chỉnh lưu hay viết tắt là ReLU là một hàm tuyến tính từng đoạn sẽ trực tiếp xuất đầu vào nếu nó là tích cực, nếu không, nó sẽ xuất ra bằng không. ReLU đã trở thành hàm kích hoạt mặc định cho nhiều loại mạng nơ-ron vì mô hình sử dụng nó dễ huấn luyện hơn và thường đạt được hiệu suất tốt hơn. Để sử dụng giảm độ dốc ngẫu nhiên với sự lan truyền ngược của lỗi để đào tạo mạng nơ-ron, cần có một hàm kích hoạt giống như một hàm tuyến tính, nhưng bản chất là một hàm phi tuyến tính cho phép học các mối quan hệ phức tạp trong dữ liệu. ReLU có thể được mô tả bằng bài toán tìm số lớn nhất như sau: $Max\,(0, x)$

$$f\,(x) = \begin{cases} 0 & for \;\; x \;<\; 0 \\ x & for \;\; x \;\geq\; 0 \end{cases} \tag{3.1}$$

- Leaky ReLU: Một trong số các vấn đề mà ReLU gặp phải chính là tồn tại nơ-ron ReLU không được kích hoạt với tất cả đầu vào, vì vậy không có độ dốc nào được tính, nếu tình trạng này xảy ra trên một số lượng lớn nơ-ron, hiệu suất của mạng nơ-ron sẽ bị ảnh hưởng. Leaky ReLU sẽ giải quyết vấn đề này, Leaky RelU để cho phép các giá trị âm nhỏ khi đầu vào nhỏ hơn 0.

$$f\,(x) = \begin{cases} 0.01x & for \;\; x \;\leq\; 0 \\ x & for \;\; x \;>\; 0 \end{cases} \tag{3.2}$$

- Softmax:Hàm softmax là một hàm biến một vectơ gồm K giá trị thực thành một vectơ gồm K giá trị thực có tổng bằng 1. Giá trị đầu vào có thể là dương, âm, không hoặc lớn hơn một, softmax sẽ biến chúng thành các giá trị giữa 0 và 1, để chúng có thể được hiểu là xác suất. Nếu một trong các đầu vào nhỏ hoặc âm, thì softmax sẽ biến nó thành một xác suất nhỏ và nếu một đầu vào lớn, thì softmax sẽ biến nó thành một xác suất lớn, nhưng giá trị sẽ luôn nằm trong khoảng

từ 0 đến 1. Hàm kích hoạt softmax thường được thêm vào thành một lớp cuối cùng của mạng nơ-ron.

$$\sigma\left(\vec{z}\right)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.3}$$

### 3.1.6 Thuật toán tối ưu

- Gradient descent:Trong máy học, các bài toán tìm giá trị lớn nhất và nhỏ nhất thường xuyên được đặt ra. Có các phương pháp khác nhau để giải bài toán này, một trong số đó là xuất phát từ một điểm mà được coi là gần với nghiệm của bài toán, sau đó dùng một phép toán lặp để tiến dần đến điểm cần tìm, tức đến khi đạo hàm gần với 0. Gradient Descent và các biến thể của nó là một trong những phương pháp được dùng nhiều nhất. Giả sử cần tìm global minimum cho hàm $f\left(\theta\right)$ trong đó $\theta$ là một vector. Đạo hàm của $f\left(\theta\right)$ tại một điểm $\theta$ bất kì là $\nabla_\theta f(\theta)$, $\theta_0$ là điểm dự đoán, sau đó, ở vòng lặp thứ t, quy tắc cập nhật là:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta f(\theta_t) \tag{3.4}$$

- Stochastic gradient descent:
  - Đây là một biến thể của gradient descent, trong thuật toán này, tại 1 thời điểm chỉ tính đạo hàm của hàm mất mát dựa trên chỉ một điểm dữ liệu $x_i$ rồi cập nhật $\theta$ dựa trên đạo hàm này. Những điểm trên là ngẫu nhiên.

- Adam: (viết tắt của Adaptive Moment Estimation) là một bản cập nhật cho trình tối ưu hóa RMSProp. Trong thuật toán tối ưu hóa này, việc chạy trung bình của cả gradient và thời điểm thứ hai của gradient đều được sử dụng. cho tham số $w^{(t)}$ và hàm chi phí $L^{(t)}$, với t là chỉ số của vòng lặp hiện tại, tham số Adam được cập nhật bởi

công thức:

$$m_w^{(t+1)} \longleftarrow \beta_1 m_w^{(t)} + (1-\beta_1)\nabla_w L^{(t)} \tag{3.5}$$

$$v_w^{(t+1)} \longleftarrow \beta_2 v_w^{(t)} + (1-\beta_2)\left(\nabla_w L^{(t)}\right)^2 \tag{3.6}$$

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1-\beta_1^{(t+1)}} \tag{3.7}$$

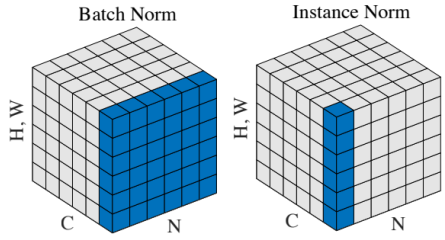
$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-\beta_2^{(t+1)}} \tag{3.8}$$

$$\hat{w}^{(t+1)} = w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_m} + \epsilon} \tag{3.9}$$

Với $\epsilon$ rất nhỏ để ngăn trường hợp chia cho số $0.\beta_1, \beta_2$ lần lượt là các yếu tố quên đối với gradient và khoảnh khắc thứ hai của gradient.

### 3.1.7 Chuẩn hóa dữ liệu (Normalization)

Một trong những phép biến đổi quan trọng thường được áp dụng vào dữ liệu đó chính là biến đổi tỉ lệ đặc trưng (scaling feature). Các thuật toán máy học thường không hoạt động tốt khi các con số của các thuộc tính có các vùng giá trị ở các tỉ lệ khác nhau. Để các thuộc tính có cùng tỉ lệ vùng giá trị thì chúng ta sử dụng phương pháp chuẩn hóa dữ liệu. Chuẩn hóa dữ liệu được tính tổng quát như sau: trừ đi giá trị trung bình của toàn bộ phần tử và sau đó chia cho độ lệch chuẩn.

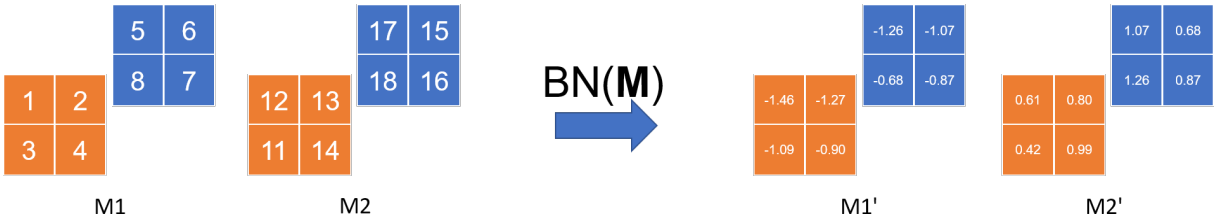



Hình 3.3: Chuẩn hóa theo batch và chuẩn hóa theo đối tượng

Trong hình 3.3, mỗi hình con là 1 tensor ánh xạ đặc trưng với N đại diện trục lô, C đại diện trục kênh và (H, W) đại diện trục không gian. Các ô pixel màu xanh được chuẩn hóa bởi cùng giá trị trung bình và phương sai, được tính bằng cách tổng hợp giá trị của các ô pixel này

- Chuẩn hóa theo lô (Batch normalization)

  Công trình chính vào năm 2015 của Ioffe và Szegedy đã giới thiệu phương pháp chuẩn hóa theo lô giúp dễ dàng cho việc huấn luyện các mạng truyền xuôi (feed-forward networks) bằng việc chuẩn hóa các đặc trưng thống kê. Chuẩn hóa theo lô được thiết kế với mục đích ban đầu là giúp tăng tốc độ huấn luyện các mạng phân biệt. Tuy nhiên nó cũng được phát hiện rằng có hiệu quả trong các mạng tạo sinh. Với ảnh đầu vào có kích thước $x \in \mathbb{R}^{N \times C \times H \times W}$, BN chuẩn hóa giá trị trung bình và độ lệch chuẩn cho mỗi đặc trưng kênh độc lập.




Hình 3.4: Chuẩn hóa theo lô (Batch norm)

Chúng ta sẽ lấy ví dụ trong hình 3.4. Đầu vào chúng ta có mảng nhiều chiều **M** và đầu ra chính là mảng **M'** có cùng kích thước với mảng đầu vào. Chúng ta sẽ lần lượt tính trung bình $\mu$ và độ lệch chuẩn $\sigma$ của toàn bộ các ô vuông màu cam thuộc M1, M2:

$\mu = \frac{1+2+3+4+12+13+11+14}{8} = 8.75, \sigma^2 = \frac{(1-\mu)^2+(2-\mu)^2+...+(14-\mu)^2}{8} = 27.8125$

Bước cuối cùng là chuẩn hóa M1, M2 bằng cách trừ lần lượt M1, M2 cho giá trị trung bình $\mu$ và chia cho độ lệch chuẩn $\sigma$ ta sẽ được các giá trị M1', M2'.

Sau đây là công thức tổng quát:

$$\mathrm{BN}(x) = \gamma \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta \tag{3.10}$$

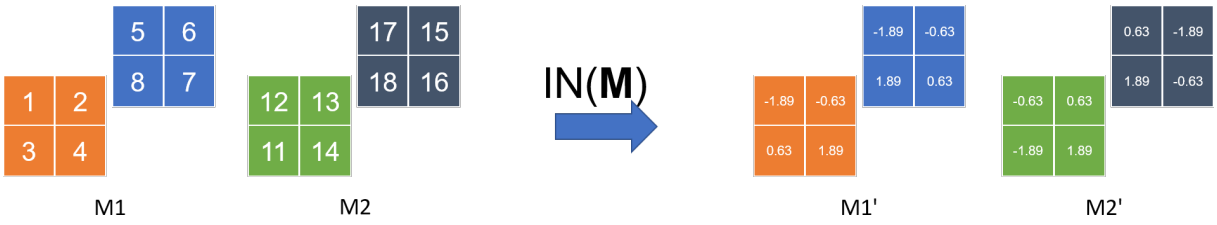
Trong đó, $\gamma, \beta \in \mathbb{R}^C$ là các tham số affine (affine parameters) học được từ dữ liệu; $\mu(x), \sigma(x) \in \mathbb{R}^C$ là giá trị trung bình và độ lệch chuẩn, được tính trên kích thước lô và không gian chiều một cách độc lập với mỗi kênh đặc trưng.

$$\mu_c(x) = \frac{1}{NHW} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} x_{nchw} \tag{3.11}$$

$$\sigma_c(x) = \sqrt{\frac{1}{NHW} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_{nchw} - \mu_c(x))^2 + \epsilon} \tag{3.12}$$

- Chuẩn hóa theo đối tượng (Instance Normalization)

  Với phương pháp cách điệu hóa truyền xuôi ban đầu, mạng biến đổi phong cách gồm một tầng BN theo sau mỗi tầng tích chập. Đáng ngạc nhiên, Ulyanov và cộng sự nhận thấy rằng có thể đạt được sự cải thiện đáng kể chỉ bằng cách thay thế các tầng BN bằng tầng IN.




Hình 3.5: Chuẩn hóa theo đối tượng (Instance norm)

$$\mathrm{IN}(x) = \gamma \left( \frac{x - \mu(x)}{\sigma(x)} \right) + \beta \tag{3.13}$$

Tuy nhiên, khác với các tầng BN, $\mu(x)$ và $\sigma(x)$ được tính trên không gian chiều độc lập với mỗi kênh và *với mỗi mẫu thử*.

$$\mu_{nc}(x) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W} x_{nchw} \tag{3.14}$$

$$\sigma_{nc}(x) = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(x_{nchw} - \mu_{nc}(x))^2 + \epsilon} \tag{3.15}$$

- Chuẩn hóa theo đối tượng thích ứng (Adaptive Instance Normalization)

  Nếu như IN chuẩn hóa đầu vào thành một kiểu duy nhất được chỉ định bởi các tham số affine, liệu rằng ta có thể chuyển đổi nó thành một kiểu bất kỳ bằng cách sử dụng các biến đổi affine thích ứng? Chuẩn hóa theo đối tượng thích ứng (AdaIN) được đề xuất như là một phần mở rộng của IN. AdaIN nhận hai đầu vào, nội dung x và kiểu y, và sau đó canh chỉnh giá trị trung bình và phương sai của x cho phù hợp với y. Không như BN và IN, AdaIN không có các tham số affine có thể học được (learnable affin parameters). Thay vào đó, nó tính toán một cách thích ứng các tham số affine từ đầu vào kiểu

$$\text{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y) \tag{3.16}$$

  Trong đó, chúng ta chỉ cần lấy tỉ lệ đầu vào nội dung đã được chuẩn hóa với $\sigma(y)$, và sau đó tịnh tiến với $\mu(y)$. Tương tự như IN, các thống kê này được tính toán trên các vị trí không gian.

  Một cách trực quan, chúng ta hãy xem xét một kênh đặc trưng có thể phát hiện nét vẽ của một kiểu nhất định. Một hình ảnh phong cách với loại nét này sẽ tạo ra kích hoạt trung bình cao cho đặc trưng này. Đầu ra do AdaIN tạo ra sẽ có cùng độ kích hoạt trung bình cao

cho đặc trưng này, đồng thời bảo toàn cấu trúc không gian của hình ảnh nội dung. Đặc trưng nét vẽ có thể được đảo ngược vào không gian hình ảnh bằng bộ giải mã truyền xuôi. Phương sai của kênh đặc trưng này có thể mã hóa thông tin kiểu tinh tế hơn, thông tin này cũng được chuyển đến đầu ra AdaIN và hình ảnh đầu ra cuối cùng.

Nói tóm lại, AdaIN thực hiện chuyển kiểu trong không gian đặc trưng bằng cách chuyển các thống kê đặc trưng, cụ thể là phương sai và trung bình phù hợp với kênh. Tầng AdaIN đóng vai trò tương tự như lớp hoán đổi kiểu được đề xuất trong [?]. Trong khi phép tính hoán đổi kiểu rất tốn thời gian và bộ nhớ, tầng AdaIN này đơn giản như một tầng IN, hầu như không thêm chi phí tính toán.

### 3.1.8 Xác suất thống kê trong bài toán máy học

- Phân phối tự nhiên (Gaussian distribution) : Phân phối này có ý nghĩa quan trọng và được áp dụng rộng rãi trong cuộc sống.Ví dụ: nếu chúng ta lấy ngẫu nhiên chiều cao của 100 người, sắp xếp từ giá trị thấp nhất đến cao nhất và vẽ thành biểu đồ với trục hoành là chiều cao, trục tung là số lượng người có chiều cao đó thì sẽ được một biểu đồ dạng hình chuông. Đây cũng chính là dạng biểu đồ của phân phối chuẩn, biểu đồ hình chuông. Về công thức, cho một biến x thuộc vào phân phối chuẩn với giá trị trung bình và độ lệch chuẩn , chúng ta sẽ có hàm mật độ phân phối xác suất sau đây:

$$P\left(X = x|\mu, \sigma^2\right) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

- Phân phối đồng nhất (Uniform distribution): Phân phối đồng nhất là một dạng phân phối trong đó tất cả kết quả có khả năng xảy ra là như nhau. Ví dụ: trong một bộ bài, tỉ lệ rút được một lá bài có hình trái tim hoặc rút được lá bài có hình kim cương là như nhau.

Việc tung một đồng xu hai mặt cũng có phân phối đồng nhất vì tỉ lệ mặt trên là mặt ngửa hay mặt sấp là như nhau. Đồ thị biểu diễn của phân phối đồng nhất sẽ là một đường thẳng nằm ngang.

- Entropy: Trong lý thuyết thông tin, entropy của một biến ngẫu nhiên là mức trung bình của "thông tin", "sự bất ngờ", hoặc "sự không chắc chắn" vốn có trong các kết quả có thể có của biến đó. Khái niệm entropy thông tin được Claude Shannon đưa ra trong bài báo "Lý thuyết toán học về truyền thông" năm 1948. Ví dụ: một đồng xu không đồng nhất (biased coin) được tung lên có xác suất mặt ngửa nằm trên là p, mặt sấp nằm trên là 1-p. Thì lúc này sự không chắc chắn lớn nhất khi $p = \frac{1}{2}$ khi không có chứng cứ nào xác định mặt trên là mặt ngửa hay mặt sấp, entropy lúc này là bit 1. Sự không chắc chắn nhỏ nhất khi p = 0 hoặc p =1 khi chúng ta đã biết chắc mặt nào của đồng xu sẽ nằm ở trên, entropy lúc này là bit 0. Khi p thay đổi sẽ dẫn đến giá trị của entropy biến đổi theo từ bit 0 đến bit 1.

  Cho một biến ngẫu nhiên rời rạc X, kết quả có thể được sinh ra là $x_1, ..., x_n$ với xác suất xảy ra là $P_{x_1}, ..., P_{x_n}$, entropy của X được thể hiện qua công thức sau:

$$H(X) = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

  Ví dụ: Hãy kể cho một người nào đó nghe về thông tin mà họ đã biết rồi, người đó sẽ không học thêm được gì cả. Vì vậy entropy trong trường hợp này sẽ rất thấp.

  Nhưng nếu kể cho người đó về những thông tin mà người đó biết rất ít, hoặc chưa biết, thì người đó sẽ có được thông tin mới. Thông tin mới này có giá trị với họ. Người đó học được điều mới. Entropy trong trường hợp này sẽ có giá trị cao.

- Cross-Entropy: cho 2 phân phối p và q, cross entropy đo lượng thông tin trung bình cần thiết để nhận dạng các mẫu ngẫu nhiên từ q khi sử dụng lược đồ mã hoá tối ưu được xây dựng cho p.

## 3.2 Lý thuyết nền tảng học máy

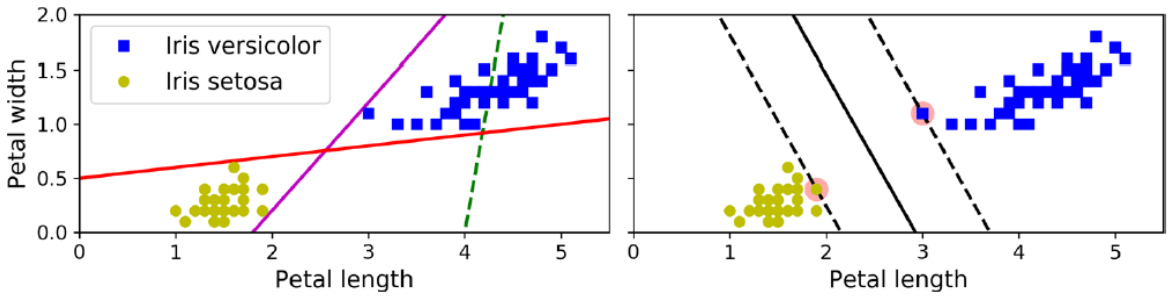
### 3.2.1 Support Vector Machine (SVM)

- Giới thiệu

SVM là một mô hình học máy mạnh mẽ và linh hoạt, có khả năng thực hiện cả phân loại tuyến tính, phân loại phi tuyến, hồi quy và thậm chí phát hiện ngoại lệ. Đây là một trong những mô hình phổ biến nhất trong Học Máy. SVM đặc biệt phù hợp để phân loại các tập dữ liệu phức tạp có kích thước vừa vả nhỏ. Trong phạm vi đề tài khóa luận, nhóm sẽ trình bày bộ phân loại SVM tuyến tính.

- Bộ phân loại SVM tuyến tính.

Chúng ta sẽ xét ví dụ trong hình ảnh sau để có hiểu được ý tưởng cốt lõi của mô hình này. Hình 3.6 cho ta thấy một phần của bộ dữ liệu iris. Hai lớp của dữ liệu này (Iris versicolor và Iris setosa) có thể dễ dàng phân tách bằng một đường thẳng (được gọi là có thể chia cắt tuyến tính). Đồ thị bên trái cho thấy ranh giới quyết định của ba bộ phân loại tuyến tính có thể có. Trong mô hình, đường ranh giới quyết định được biểu diễn bằng đường đứt nét tệ đến mức nó thậm chí không phân tách các lớp một cách chính xác. Hai mô hình khác hoạt động hoàn hảo trên tập huấn luyện này, nhưng đường ranh giới quyết định của chúng đến gần với các đối tượng mà các mô hình này có thể sẽ không hoạt động tốt trên các đối tượng mới. Ngược lại, đường liền nét trong đồ thị bên phải thể hiện ranh giới quyết định của bộ phân loại SVM. Đường này không chỉ phân tách hai lớp mà còn nằm càng xa các đối tượng huấn luyện gần nhất càng tốt. Chúng ta có thể nghĩ về một bộ phân loại SVM tối ưu khi con đường rộng nhất có thể (được biểu thị bằng các đường đứt nét song song) giữa các lớp.




Hình 3.6: Bộ phân loại có lề lớn

- Cấu tạo bên trong

### 3.2.2 Ensemble Learning

Giả sử chúng ta đặt ra một câu hỏi phức tạp cho hàng nghìn người ngẫu nhiên. Sau đó tổng hợp câu trả lời của họ. Trong nhiều trường hợp, chúng ta sẽ thấy rằng câu trả lời tổng hợp này tốt hơn câu trả lời đến từ một chuyên gia. Đây được gọi là sự sáng suốt của đám đông. Tương tự, nếu chúng ta tổng hợp các dự đoán của một nhóm các bộ dự đoán (chẳng hạn như bộ phân loại hoặc bộ hồi quy), chúng ta thường sẽ nhận được các dự đoán tốt hơn so với dự đoán riêng lẻ tốt nhất. Một nhóm các bộ dự đoán được gọi là *ensemble*. Do đó, kỹ thuật này được gọi là *Ensemble Learning*, và thuật toán Ensemble Learning được gọi là *phương pháp Ensemble*.

Có nhiều phương pháp Ensemble khác nhau. Phổ biến nhất hiện nay thì chúng ta có các phương pháp như: *bagging*, *boosting*, và *stacking*. Trong phạm vi của luận văn, nhóm sẽ trình bày về phương pháp boosting.

- Phương pháp boosting

Boosting đề cập đến bất kỳ phương pháp Ensemble nào có thể kết hợp một số bộ học yếu thành một bộ học mạnh. Ý tưởng chung của hầu hết các phương pháp Boosting là huấn luyện các bộ dự đoán một cách tuần tự, mỗi bộ cố gắng sửa chữa, rút kinh nghiệm từ bộ tiền nhiệm của nó. Nhóm sẽ lấy phương pháp Gradient Boosting làm phương pháp trình bày cụ thể trong số các phương pháp Boosting hiện có.

- Gradient Boosting

Đây là một phương pháp rất nổi tiếng trong các phương pháp Boosting. Ý tưởng phương pháp này như sau: lần lượt thêm các bộ dự đoán vào một nhóm ensemble, mỗi bộ sẽ tiếp tục hoàn chỉnh bộ tiền nhiệm của nó. Với mỗi bộ dự đoán mới được thêm vào nhóm ensemble, đầu vào của nó sẽ là *các lỗi phần dư* (residual errors) của bộ tiền nhiệm. 20