Báo cáo mô hình sinh
MADE - Masked autoencoder for distribution estimation

Giáo viên hướng dẫn: PGS.TS Thân Quang Khoát

Sinh viên thực hiện: Nguyễn Trần Khang, Phạm Văn Hoàng

Hà Nội- Tháng 10 năm 2020

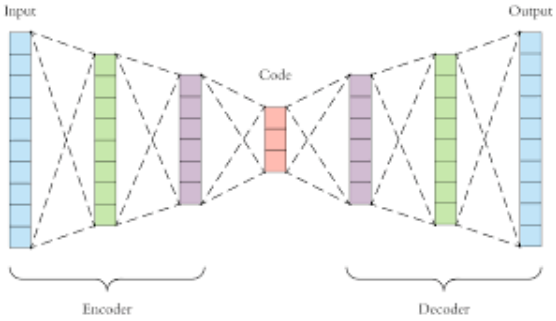
Tóm tắt nội dung

Các mô hình sinh được sử dụng rộng rãi trong nhiều lĩnh vực về trí tuệ nhân tạo (AI) và học máy (Machine learning). Các phương pháp tiếp cận hiện tại chủ yếu xoay quanh các mô hình sử dụng mạng nơ ron, kết hợp với sự phát triển của các phương pháp tính toán tối ưu, từ đó cho ra được các mô hình bao phủ được các loại dữ liệu nhiều chiều như ảnh, văn bản và tiếng nói. Trong quá trình tìm hiểu các lớp mô hình như vậy, nhóm nghiên cứu quyết định chọn mô hình MADE- Masked Autoencoder for Distribution Estimation để tiến hành nghiên cứu, cài đặt, thử nghiệm và báo cáo

1 Giới thiệu

Mô hình MADE được phát triển chỉnh sửa ý tưởng từ mô hình Autoencoder đơn giản trước đó.

Ý tưởng của Autoencoder khá đơn giản: Nhận một đầu vào là x và chuyển về một dạng biến ẩn z mang một đặc trưng nào đó của đầu vào (Encoder); sau đó tiến hành tái tạo z trở về x như ban đầu (Decoder). Trong khuôn khổ của báo cáo, nhóm xét tới xử lí các bài toán sinh ảnh đen trắng.



Hình 1: Vanilla AutoEncoder

Mặc dù mô hình Vanilla Autoencoder có thể học được một dạng biến ẩn cho dữ liệu, tuy vậy mô hình không có tính giải thích xác suất. Có nghĩa rằng là với mọi giá trị của dữ liệu, tổng xác suất của các khả năng xảy ra phải bằng 1. $\sum_x P(x) = 1$

2 Mô hình MADE

2.1 Tính tự hồi quy(Autoregressive)

Để khắc phục nhược điểm của mô hình truyền thống, mô hình MADE ra đời. Điểm mấu chốt trong mô hình đó là các Input được tái tạo từ các Input trước nó theo một thứ tự chọn trước. Đầu ra của mô hình Autoregressive là các phân phối xác suất có điều kiện, thay vì là bản tái tạo của input. Cách tiếp cận này hợp lý, giải thích được bởi nhiều tổ hợp xác suất có điều kiện.
Ước lượng phân phối (Distribution Estimation) là quá trình ước lượng một xác suất hợp $p(x)$ từ một tập các mẫu $\{x^{(t)}\}$. Đầu tiên, với một phân phối bất kì, xác suất đồng thời luôn được với về dạng tích của các xác suất có điều kiện.

$$p(x) = \prod_{d=1}^{D} p(x_{\mathrm{d}}|x_{<\mathrm{d}}) \tag{1}$$

trong đó

$$x_{<\mathrm{d}} = \{x_1, ..., x_{\mathrm{d}\text{-}1}\}$$

Như vậy mỗi output $\hat{x_{\mathrm{d}}} = p(x_{\mathrm{d}}|x_{<\mathrm{d}})$ phải phụ thuộc vào input $x_{<\mathrm{d}}$ trước đó, output của thời điểm hiện tại sẽ chỉ phuộc thuộc vào các input trước đó, đặc trưng của các mô hình Autoregressive.

2.2 Cơ chế mặt nạ Autoencoders(Masked Autoencoders)

Câu hỏi bây giờ là chỉnh sửa mô hình Autoencoder sao cho thỏa mãn đặc trưng của mô hình Autoregressive.

Do output $\hat{x_d}$ chỉ phụ thuộc vào các input $x_{<d}$, như vậy trong mạng nơ rơn sẽ không có đường đi từ các $x_{>d}$ tới $\hat{x_d}$, nói cách khác trọng số ứng với các đường làm mất đi tính Autoregressive sẽ phải bằng 0.
Hướng tiếp cận của MADE sử dụng nhân các bộ trong số của từng lớp (Elementwise) với ma trận nhị phân gọi là mặt nạ (Mask matrix). Các phần tử của Mask matrix nhận giá trị $\{0, 1\}$, ứng với việc che hay không che các trọng số của mạng.

$$h(x) = g(b + (W \odot M^{\mathrm{W}})X) \tag{2}$$

$$\hat{x} = sigm(c + (V \odot M^{\mathrm{V}})h(x)) \tag{3}$$

trong đó:

- $\odot$ là toán tử nhân từng phần tử
- $x, \hat{x}$ là vector input/output tương ứng
- $h(x), g()$ là lớp ẩn và hàm kích hoạt tương ứng
- $sigm(.)$ là hàm sigmoid đầu ra ứng với bài toán ảnh đen trắng
- b, c là các nhiễu
- W, V là ma trận trọng số
- $M^{\mathrm{W}}, M^{\mathrm{V}}$ là Mask matrix tương ứng

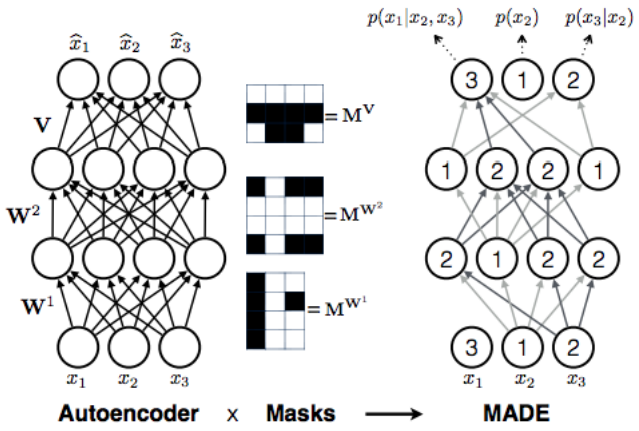
Như vậy chỉ còn một điều kiện nữa để giả thiết đã được thỏa mãn, mạng sẽ có thể phù hợp với các phân phối xác suất. Bước tiếp theo là việc chọn Masks matrix như thế nào cho hợp lý. Với mỗi thứ tự cho trước của các input sẽ có rất nhiều kiểu Mask để đảm bảo. Một cách tổng quát, với mỗi node ở lớp ẩn, ta định một chỉ số quyết định xem node này sẽ được kết nối với inputs và outputs như thế nào. Gọi $m^{\mathrm{l}}(k)$ là chỉ số gán cho node thứ k trong lớp thứ l. $m^{\mathrm{l}}(k)$ được lấy ngẫu nhiên theo phân phối đều trong khoảng $[1, D-1]$ với D là số chiều dữ liệu. Mask matrix của lớp ẩn đầu tiên như sau.

$$M^{W^{\mathrm{l}}}_{k',k} = \begin{cases} 1 \text{ if } m^l(k') \geq m^{l-1}(k) \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

Đối với lớp output, Mask matrix có dạng gần giống.

$$M^{V}_{d,k} = \begin{cases} 1 \text{ if } d > m^L(k) \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

.



Hình 2: Autoencoder to MADE

2.3 Quá trình huấn luyện và sinh

Quá trình huấn luyện nhận Input là một ảnh x và output là các xác suất có điều kiện $p(x_{\mathrm{d}}|x_{<\mathrm{d}})$. Trong bài toán ta đang xét, do ảnh là ảnh đen trắng, nên mỗi xác suất đầu ra đầu ra là một tham số $p = \hat{x}_{\mathrm{d}}$ của phân phối Bernoulli. Hàm mục tiêu cho quá trình huấn luyện dựa trên cực đại hóa hàm Likelihood (Maximum Likelihood Estimation-MLE).

$$\operatorname{argmax} \log(P(x)) \tag{6}$$

$$\operatorname{argmax} \sum_{d} \log(P(x_{\mathrm{d}}|x_{<\mathrm{d}})) \tag{7}$$

Pha huấn luyện sẽ thực hiện song song giữa các x_{d}. Tối ưu theo chiến lược Stochastic Gradient Descent (SGD).
Quá trình sinh khác với quá trình huấn luyện. Mỗi x_{d} sẽ được sinh lần lượt nhau và dùng kết quả sinh được ở các lần trước đó để sinh tiếp pixel tiếp theo. Do đó với dữ liệu D chiều, mỗi lần sinh ảnh chúng ta phải đưa qua mạng D lần, mỗi lần tiến hành tính toán xác suất đầu ra, lấy mẫu từ xác suất đầu ra và đưa kết quả làm Input cho lần tiếp theo. Quá trình sinh diễn ra như thuật toán sau.

B1 Chọn vector x bất kì D chiều, d=1

B2 Chọn thứ tự sinh

B3 Tạo Mask

B4 Đưa x qua mạng và nhận output $\hat{x_{\mathrm{d}}}$

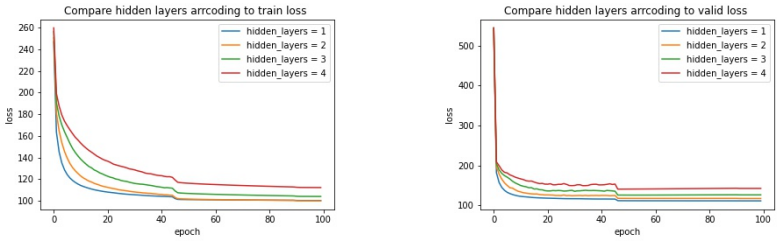
B5 Lấy mẫu Bernoulli với $p = \hat{x_{\mathrm{d}}}$. Gán giá trị cho xd và tăng d 1 đơn vị. Lặp lại B4,B5 cho tới khi d = D.

3 Cài đặt, kiểm thử

3.1 Dữ liệu

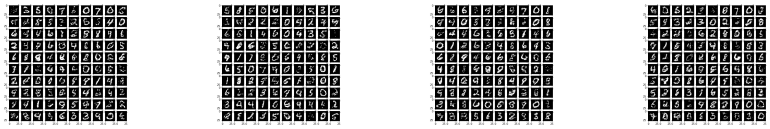
Mô hình MADE sẽ được cài đặt, kiểm thử trên 3 bộ dữ liệu MNIST chữ số, EMNIST chữ cái và FOUR SHAPE. Số lượng ảnh (train, test) tương ứng cho ba bộ. (50000, 10000), (124800, 20800), (1600, 400). Cả 2 bộ dữ liệu đều là tập các ảnh đen trắng về chữ số và chữ cái. Số chiều một ảnh đều là 28x28, mỗi pixel nhận giá trị 0 hoặc 1. Do đó đầu ra của bài toán được chọn là phân phối Bernoulli.

3.2 Đánh giá ảnh hưởng độ sâu mô hình



(a) train loss (b) test loss

Hình 3: Hàm mất mát theo độ sâu mô hình



(a) 1 layers (b) 2 layers (c) 3 layers (d) 4 layers

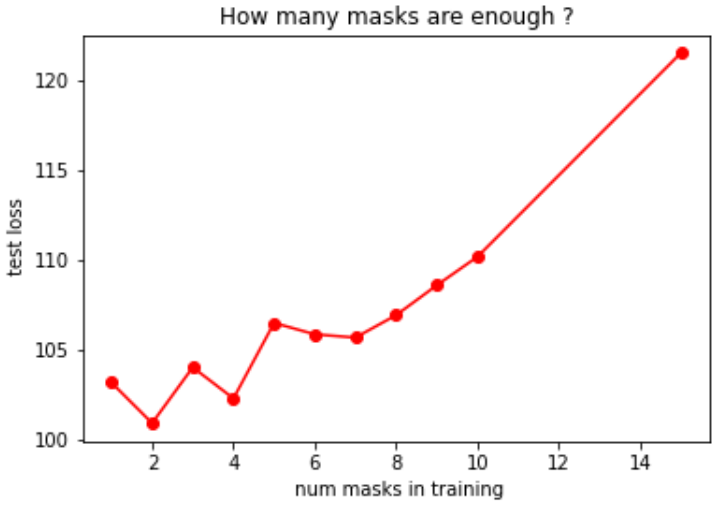
Hình 4: Ảnh sinh ra theo độ sâu mô hình

Theo thực nghiệm với bộ dữ liệu MNIST, với độ sâu bằng 1 sẽ cho tốc độ hội tụ tốt hơn ở cả trên tập huấn luyện và kiểm thử. Tuy nhiên không có khác biệt quá lớn ở việc sinh ảnh.

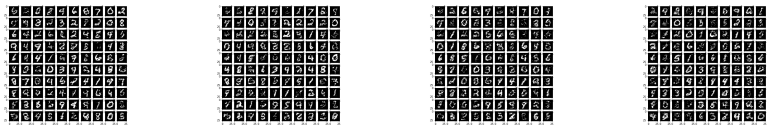
3.3 Đánh giá ảnh hưởng số lượng Mask

Trong quá trình huấn luyện, sau một vài minibatch, ta sẽ cập nhật lại masks để huấn luyện. Số masks được sử dụng theo cách xoay vòng trong một tập hợp masks đã tạo. Quá trình kiểm thử cho thấy việc sử dụng số lượng masks

quá nhiều sẽ ảnh hưởng đến chất lượng của mô hình. Hiện tượng này gọi là Overregularization.



Hình 5: Số lượng Masks ảnh hưởng lên hàm mất mát.

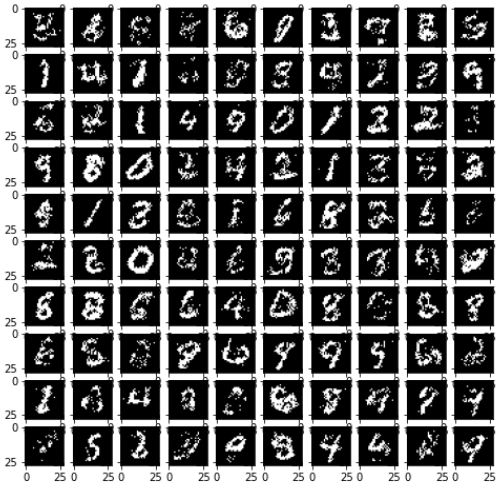


(a) 2 masks (b) 4 masks (c) 10 masks (d) 15 masks

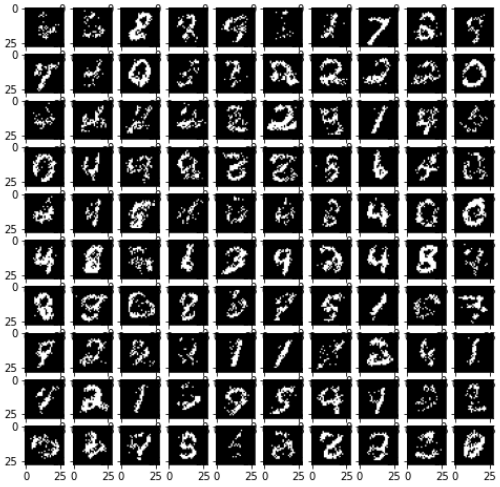
Hình 6: Ảnh sinh ra theo số lượng mask sử dụng.

3.4 Đánh giá cách sinh ảnh mới

Nhóm có thực hiện kiểm thử khả năng sinh ảnh theo một cách khác. Thay thế cho chiến lược mỗi lần sinh chọn một masks. Ở mỗi lần sinh, tiến hành tái tạo nhiều masks sử dụng đưa qua mạng rồi lấy kết quả trung bình để lấy mẫu. Kết quả khi sinh ảnh cho thấy cả hai cách đều cho kết quả tương tự nhau.



(a) Dùng nhiều masks(mới)

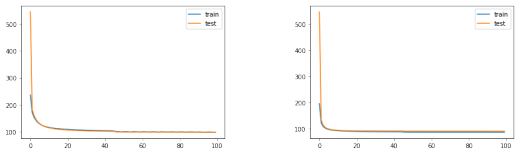


(b) Dùng đơn masks(gốc)

Hình 7: Ảnh theo 2 cách sinh.

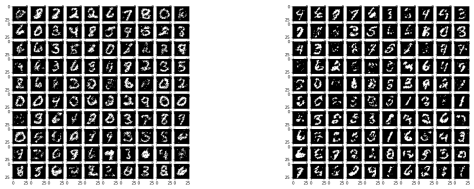
3.5 Đánh giá ảnh hưởng của kết nối trực tiếp (skip connection)

Skip connection kết nối trực tiếp Input với Ouput. Mô hình sử dụng skip connection cho kết quả hàm loss tốt hơn so với ban đầu, cộng thêm tốc độ hội tụ nhanh hơn. Tuy nhiên kết quả sinh ảnh so với không dùng skip connection thì tệ hơn.



(a) Không skip (b) Có skip

Hình 8: Hội tụ hàm loss



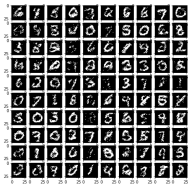
(a) Không skip (b) Có skip

Hình 9: Ảnh sinh theo 2 cách

3.6 Đánh giá ảnh hưởng của thứ tự

Thực hiện huấn luyện nhiều lần trên các thứ tự khác nhau, ta có thể thu được kết quả như trên. Với các thứ tự tự nhiên như Raster-scan, Columns cho kết quả tốt hơn. Thứ tự ngẫu nhiên sẽ làm cho mô hình không có khả năng sinh.

(a) Raster-scan



(b) Columns

(c) Randoms

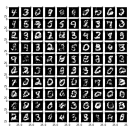
(d) ReverseRS

(e) TopMidDown

Hình 10: Ảnh sinh theo các thứ tự khác nhau

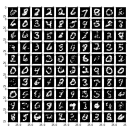
3.7 Một số kết quả sinh

(a) Epochs 0



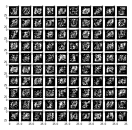
(b) Epochs 20

(c) Epochs 50



(d) Epochs 100

Hình 11: Ảnh sinh theo bộ MNIST



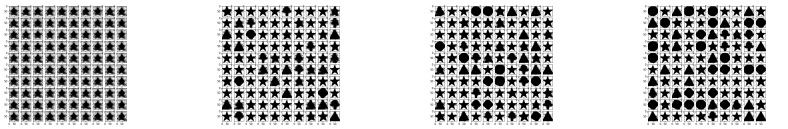
(a) Epochs 0

(b) Epochs 20

(c) Epochs 50

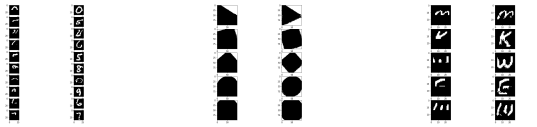
(d) Epochs 100

Hình 12: Ảnh sinh theo bộ EMNIST



(a) Epochs 0 (b) Epochs 20 (c) Epochs 50 (d) Epochs 100

Hình 13: Ảnh sinh theo bộ FOUR SHAPE



(a) MNIST (b) FOUR SHAPES (c) EMNIST

Hình 14: Khả năng tái tạo ảnh

4 Kết luận

Trong bài báo cáo này, nhóm đã tiến hành chạy thử nghiệm, đánh giá trên các bộ ảnh đen trắng với số chiều nhỏ. Kết quả cho thấy khả năng sinh ảnh đối với các bộ dữ liệu là khá tốt ở FOUR SHAPE, tương đối ở MNIST và EMNIST. Trong tương lai có thể áp dụng mô hình để kiểm thử trên ảnh xám.

Tài liệu

[1] Mathieu Germain, Karol Gregor, Iain Murray, Hugo Larochelle. *MADE: Masked Autoencoder for Distribution Estimatio*

[2] Brian Keng *http://bjlkeng.github.io/posts/autoregressive-autoencoders/*

[3] Andrej Karpathy *https://github.com/karpathy/pytorch-made/*

[4] CS294-158-SP20 Deep Unsupervised Learning Spring 2020 *https://sites.google.com/view/berkeley-cs294-158-sp20/home*