TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC QUỐC GIA TP.HCM
BỘ MÔN VIỄN THÔNG

Đề cương

Tạo ảnh từ văn bản

GVHD: ThS Nguyễn Khánh Lợi (nkloi@hcmut.edu.vn)
Sinh viên: Phan Nguyên Trung - 1814519

Hồ Chí Minh, 12/2021

Lời cảm ơn

Đầu tiên, em xin bày tỏ lòng biết ơn đến chân thành sâu sắc tới giáo viên hướng dẫn của em, thầy **ThS Nguyễn Khánh Lợi** - "giảng viên bộ môn Viễn Thông" đã trực tiếp giúp đỡ, hướng dẫn em hoàn thành đề cương này.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, bài báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn.

*Tp. Hồ Chí Minh, tháng **12** năm **2021**.*

Sinh viên

Tóm tắt đề cương

Tạo ảnh từ đoạn văn bản là một trong những đề tài thú vị trong lĩnh vực Deep Learning. Với sự phát triển nhiều kiến trúc mạng mới, nhiều phương án xử lý bài toán tạo ảnh từ văn bản đã được giới thiệu và cho ra kết quả khả quan.

Đề cương với mục tiêu nghiên cứu, thiết kế một mô hình tạo ảnh về các loài chim từ văn bản . Đồng thời tìm hiểu các model mạng sinh đối nghịch (GAN), các kiến thức về natural language processing (NLP). Thử nghiệm một số hàm tối ưu cho bài toán .

Mục lục

Danh sách hình

Chương 1

Giới thiệu

1.1 Tổng quan

Tạo hình ảnh có độ phân giải cao từ mô tả văn bản rất quan trọng đối với nhiều ứng dụng thực tế như tạo ảnh nghệ thuật và thiết kế đồ họa máy tính. Đây là một đề tài khá phức tạp vì ta phải làm việc với dữ liệu dưới dạng văn bản (chuỗi, câu, từ) và liệu dưới dạng hình ảnh. Công việc này có thể hiểu đơn giản là việc tạo ra những bức ảnh thể hiện được những đặc trưng được mô tả trong đoạn văn bản. Trong đề cương này, mỗi mẫu dữ liệu văn bản có dạng là một câu đơn tiếng Anh được con người trình bày và nó mô tả những tính chất đặc điểm của bức ảnh.

Hình 1.1: Ví dụ ảnh được tạo từ đoạn văn bản.

Việc trích xuất các đặc trưng từ đoạn văn bản mô tả các tính chất của ảnh (ảnh được tạo) là một công việc khá phức tạp. Tuy nhiên sự phát triển của các mạng Recurrent netowrks (RNN) gần đây đã giúp chúng ta dễ dàng hơn thực hiện điều này. Vấn đề khó nhất của đề tài

là tìm một phép biến đổi để ánh xạ các tính xuất đã lấy từ các đoạn văn bản thành một hình ảnh với nội dung phù hợp với các tính chất đấy.Trong thực tế, có rất nhiều cấu hình hợp lý cho các pixel của ảnh mà chúng đều có thể với đoạn văn bản được cho Nghĩa là với một nội dung của đoạn văn bản có thể sinh ra được rất nhiều ảnh khác nhau .Do đó việc tạo ảnh từ điều kiện của văn bản rất đa phương thức.

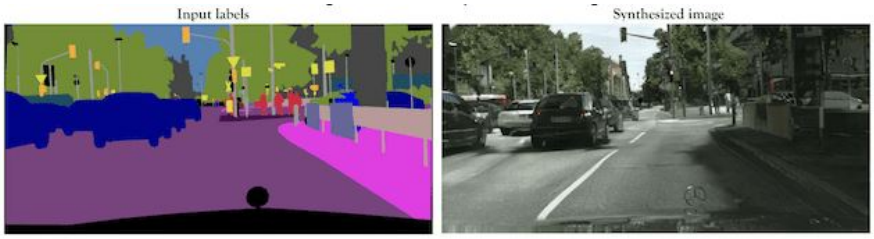
1.2 Mục tiêu

Đề cương này được đưa ra với mục đích xây dựng và huấn luyện một mô hình mạng học sâu để tạo ảnh từ đoạn văn bản. Ảnh được tạo ra phải là ảnh có mức độ chân thực tốt và minh họa đúng với nội dung của đoạn văn bản.

Chương 2

Mạng đối nghịch tạo sinh GAN

Trong những năm gần đây, một trong những xu hướng nghiên cứu thu hút được đông đảo các nhà khoa học trong lĩnh vực Deep Learning đó là mạng đối nghịch tạo sinh GAN. Vì tính ứng dụng rất cao của GAN trong các đề tài thực tế như: Tạo ảnh khuôn mặt người, tạo ảnh các vật thể, chuyển từ ảnh Semantic sang ảnh thật,..



Hình 2.1: Mô hình GAN chuyển từ ảnh Semantic sang ảnh thật .

2.1 Khải quát về mạng GAN

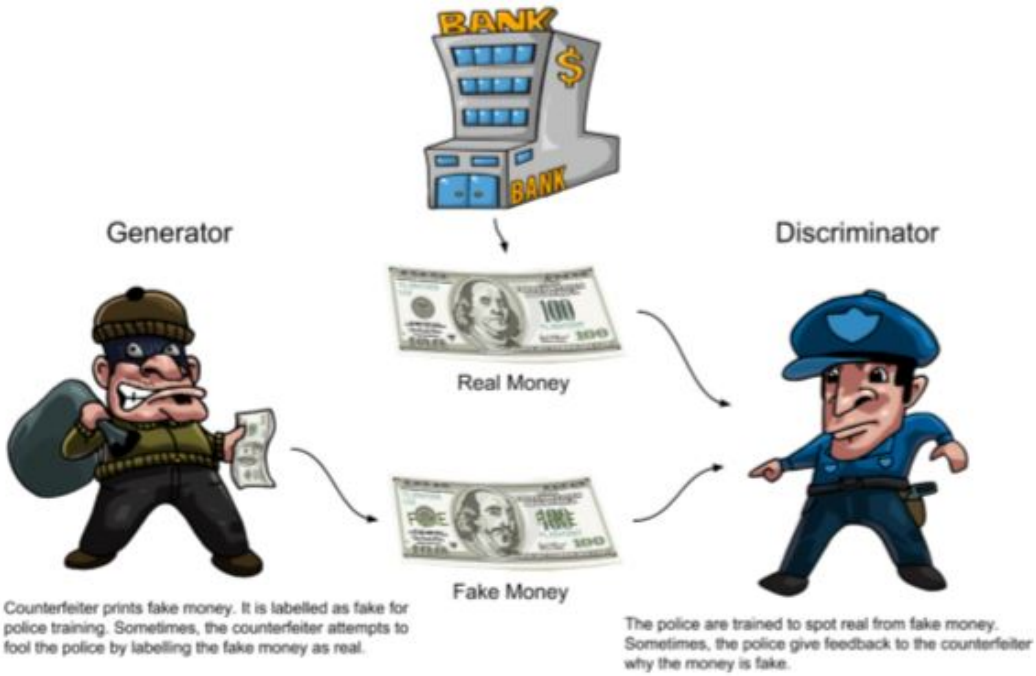
Mạng GAN thuộc nhóm mô hình sinh dữ liệu mới [2]. Dữ liệu sinh ra nhìn như thật nhưng không phải thật. Ví dụ như ảnh mặt người (hình 2.1) là do GAN sinh ra, không phải mặt người thật.

G - **G**enerative ý chỉ sinh, N - **N**etwork là mạng, còn A - **A**dversarial là đối nghịch. Lí do đối nghịch là trong mạng này được tạo nên từ sự kết hợp giữa 2 mạng là **bộ sinh** (*G* - ***G****enerator*) và **bộ phân biệt** (*D* - ***D****iscriminator*), luôn luôn đối nghịch nhau trong quá trình huấn luyện. Trong khi bộ sinh cố gắng sinh ra các dữ liệu giống như thật thì bộ phân biệt lại cố gắng phân biệt đâu là dữ liệu được sinh ra từ bộ sinh và đâu là dữ liệu thật.

Giả sử như bài toán đưa cho GAN là sinh ra tiền giả giống như tiền thật để có thể dùng được (hình 2.3), thì bộ sinh là người làm tiền giả, còn bộ phân biệt giống như cảnh sát. Người làm tiền giả sẽ cố gắng làm ra những tờ tiền giả làm sao để cảnh sát không biết đó là giả, còn cảnh sát thì cố gắng học để phân biệt được tiền nào là giả, tiền nào là thật.

Mục tiêu cuối cùng của GAN là người làm tiền giả phải có khả năng làm tiền giả, sao cho cảnh sát không phân biệt được đâu là thật đâu là giả (50/50) để đem tiền giả đi tiêu thụ. Trong quá trình huấn luyện mạng GAN, thì nhiệm vụ của cảnh sát là học cách phân biệt tiền giả và tiền thật, bên cạnh đó là nói cho người làm tiền giả là nên làm giả như thế nào cho giống thật

Hình 2.2: Ảnh mặt người sinh ra bởi GAN.



Hình 2.3: Minh hoạ bộ sinh và bộ phân biệt trong mạng GAN.

hơn. Dần dần thì người làm tiền giả sẽ làm ra được tiền giống tiền thật và cảnh sát cũng trở nên thành thạo trong việc phân biệt tiền thật hay giả.

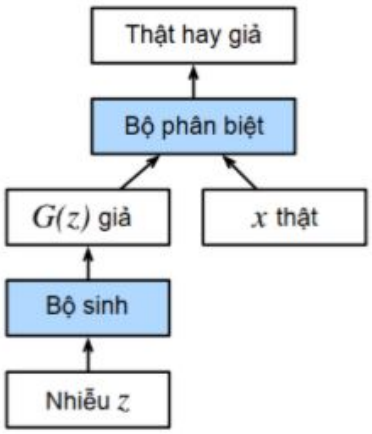
Ý tưởng của GAN bắt nguồn từ Non-Zero-Sum Games[1], là một trò chơi đối kháng giữa 2 người. Nếu một trong hai người thắng, thì người còn lại sẽ thua. Ở mỗi lượt, thì cả 2 đều muốn tối đa hoá cơ hội thắng của mình và tối thiểu hoá cơ hội thắng của đối phương. Trong

[1]Stanford, 'Non-Zero-Sum Games', https://stanford.io/3nCiLKq

lý thuyết trò chơi thì mô hình sẽ hội tụ khi cả bộ sinh và bộ phân biệt đạt tới trạng thái cân bằng Nash[2], tức là các bước tiếp theo của bất cứ ai trong hai người đều không làm thay đổi cơ hội thắng của ai cả.

2.2 Cở sở toán học và hàm mục tiêu (mất mát) của mạng GAN

Dưới góc độ toán học, một bộ sinh sẽ sinh ra dữ liệu tốt (dữ liệu mà chúng ta mong muốn) nếu ta không thể chỉ ra đâu là dữ liệu giả và đâu là dữ liệu thật. Trong thống kê, điều này được gọi là bài kiểm tra từ hai tập mẫu - một bài kiểm tra để trả lời câu hỏi liệu tập dữ liệu $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$ và $\boldsymbol{X} = \{x'_1, x'_2, \ldots, x'_n\}$ có được rút ra từ cùng một phân phối hay không. Sự khác biệt chính giữa hầu hết những bài nghiên cứu thống kê và GAN, là GAN sử dụng ý tưởng này theo kiểu có tính xây dựng. Nói cách khác, thay vì chỉ huấn luyện mô hình để nói "này, hai tập dữ liệu đó có vẻ như không đến từ cùng một phân phối", thì GAN sử dụng phương pháp **kiểm tra trên hai tập mẫu**[3] để cung cấp tín hiệu cho việc huấn luyện cho bộ sinh. Điều này cho phép ta cải thiện bộ sinh để có thể sinh dữ liệu tới khi ra được thứ gì đó giống như dữ liệu thực. Ở mức tối thiểu nhất, bộ sinh cần phải lừa được bộ phân biệt, kể cả bộ phân biệt của ta là một mạng nơ-ron sâu tân tiên nhất.



Hình 2.4: Kiến trúc mạng đối nghịch tạo sinh.

Bộ phân biệt $D(\cdot)$ là một mạng học sâu phân loại nhị phân nhằm phân biệt đầu vào $\mathbf{x} \in \mathbb{R}^n$ là thật (từ dữ liệu thật) hoặc giả (từ bộ sinh) (hình 2.4), được học theo kiểu giám sát. Đầu ra của bộ phân biệt là một số vô hướng $o \in \mathbb{R}$ dự đoán cho đầu vào $\mathbf{x}$, và sẽ được đưa qua hàm sigmoid $S(x) = 1/(1 + e^{-x}), \boldsymbol{R} = (0, 1)$ để nhận được xác suất dự đoán với giá trị càng gần 1 thì bộ phân biệt càng có xu hướng quyết định đó là dữ liệu thật. Giả sử mỗi cặp dữ liệu huấn luyện thứ i có dạng $(\mathbf{x}_i, y_i) \in \mathbb{R}^n \times \{0, 1\}$. Bộ phân biệt cần được tối ưu sẽ có dạng entropy chéo [7], nghĩa là:

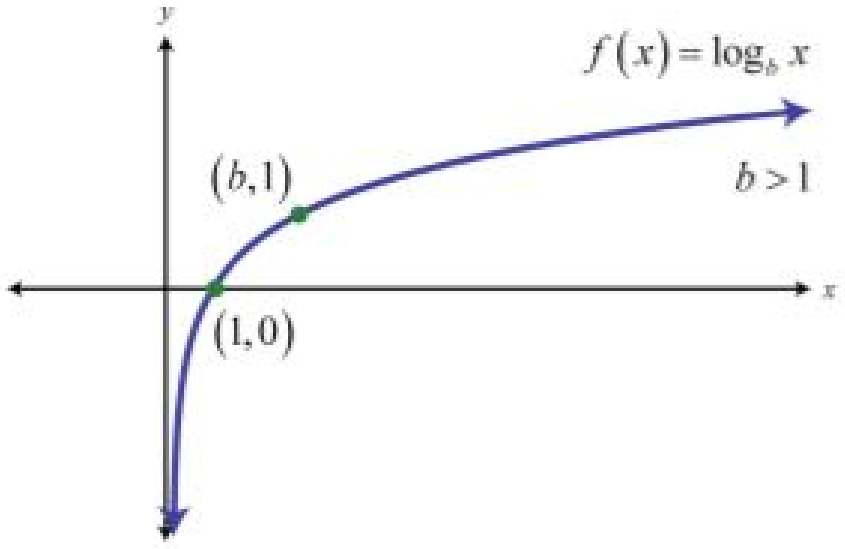
$$D^* = \arg\min_D \left\{ -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log D(\mathbf{x}_i) + (1 - y_i) \log (1 - D(\mathbf{x}_i)) \right] \right\} \tag{2.1}$$

[2]Jørgen Veisdal, 'The Nash equilibrium, explained', https://bit.ly/3ea57Lr
[3]Wikipedia, 'Two-sample hypothesis testing', https://bit.ly/3vmFwVD

Còn đối với bộ sinh $G(\cdot)$ - cũng là một mạng học sâu, sẽ được học theo kiểu không giám sát. Trước tiên nó cần được cho vài tham số ngẫu nhiên được xem là nhiễu $\mathbf{z} \in \mathbb{R}^d$ từ một nguồn[4], ví dụ phân phối chuẩn $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2) = \mathcal{N}(0, 1)$. Ta thường gọi $\mathbf{z}$ như là một biến tiềm ẩn. Mục tiêu của mạng sinh là đánh đánh lựa bộ phân biệt để phân loại $\mathbf{x}' = G(\mathbf{z})$ là dữ liệu thật, nghĩa là, ta muốn $D(G(\mathbf{z})) \approx 1$. Một cách diễn đạt khác, cho trước một bộ phân biệt D, ta sẽ cập nhật tham số của bộ sinh G nhằm cực đại hoá mất mát entropy chéo như trong (2.1) khi $y = 0$, tức là:

$$\begin{aligned} G^* &= \arg\max_G \left\{ -\frac{1}{N_{\text{nhiễu}}} \sum_{i=1}^{N_{\text{nhiễu}}} (1 - y_i) \log\left(1 - D\left(G\left(\mathbf{z}_i\right)\right)\right) \right\} \\ &= \arg\max_G \left\{ -\frac{1}{N_{\text{nhiễu}}} \sum_{i=1}^{N_{\text{nhiễu}}} \log\left(1 - D\left(G\left(\mathbf{z}_i\right)\right)\right) \right\} \end{aligned} \tag{2.2}$$

Trong thực tế, công thức (2.2) không phải là một công thức tốt để tối ưu G [6]. Là bởi vì, khi mới bắt đầu huấn luyện, bộ sinh G gần như không có khả năng sinh được dữ liệu có phân phối gần giống với phân phối ta cần. Khả năng cao là ta sẽ có $D(G(\mathbf{z})) \approx 0 \Leftrightarrow (1 - D(G(\mathbf{z}))) \approx 1$. Mà ta dễ dàng thấy, độ dốc hay chính là đạo hàm của hàm $\log(x)$ giảm khi $x : 0 \to 1$ (hình 2.5, xét cơ số $b = e$).



Hình 2.5: Đồ thị hàm $\log_b x$.

Nếu như x gần giá trị 1, đạo hàm của $\log(x)$ sẽ bé. Dễ gây hiện tượng biến mất gradient [8], không cập nhật tham số được cho G. Thay vào đó, khi tối ưu G, ta sẽ sử dụng chiến lược

[4]Số chiều d của nhiễu đầu vào cũng có ảnh hưởng tới kết quả của mô hình GAN. Và không có một con số nào là tuyệt đối tốt nhất cho mọi kiến trúc [4].

cực tiểu hoá mất mát entropy chéo như trong (2.1) với $y = 1$:

$$\begin{aligned} G^* &= \arg\min_G \left\{ -\frac{1}{N_{\text{nhiễu}}} \sum_{i=1}^{N_{\text{nhiễu}}} y_i \log\left(D\left(G\left(\mathbf{z}_i\right)\right)\right) \right\} \\ &= \arg\min_G \left\{ -\frac{1}{N_{\text{nhiễu}}} \sum_{i=1}^{N_{\text{nhiễu}}} \log\left(D\left(G\left(\mathbf{z}_i\right)\right)\right) \right\} \end{aligned} \tag{2.3}$$

Khi làm như vậy, ta sẽ có đạo hàm của $\log\left(D\left(G\left(\mathbf{z}\right)\right)\right)$ tại giá trị $D\left(G\left(\mathbf{z}\right)\right) \approx 0$ là lớn hơn so với (2.2), tránh va phải việc gradient biến mất. Vì đạo hàm của $\log(x)$ khi x ở gần 0 lớn hơn khi x ở gần 1.

Tóm lại, D và G đang chơi trò "cực tiểu hoá cực đại" với một hàm mục tiêu toàn diện như sau:

$$(D, G)^* = \arg\min_D \arg\max_G \left\{ -\mathbb{E}_{\mathbf{x}\sim\text{dữ liệu thật}} \log D\left(\mathbf{x}\right) - \mathbb{E}_{\mathbf{z}\sim\text{nhiễu}} \log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right) \right\} \tag{2.4}$$

$$\Leftrightarrow (D, G)^* = \arg\min_G \arg\max_D \left\{ \mathbb{E}_{\mathbf{x}\sim\text{dữ liệu thật}} \log D\left(\mathbf{x}\right) + \mathbb{E}_{\mathbf{z}\sim\text{nhiễu}} \log\left(1 - D\left(G\left(\mathbf{z}\right)\right)\right) \right\} \tag{2.5}$$

Phép toán $\mathbb{E}\{\cdot\}$ trong [(2.4), (2.5)] chính là kì vọng toán, tương đương việc lấy trung bình của tất cả dữ liệu.

Từ hàm mất mát (2.5), ta nhận thấy rằng việc huấn luyện bộ phân biệt và bộ sinh là đối nghịch nhau. Trong khi D cố gắng cực đại mất mát thì G lại đi theo hướng cực tiểu mất mát. Về mặt lý thuyết, quá trình dạy học cho GAN kết thúc khi mô hình GAN đạt đến trạng thái cân bằng của hai bộ trong mạng, tức cân bằng Nash.

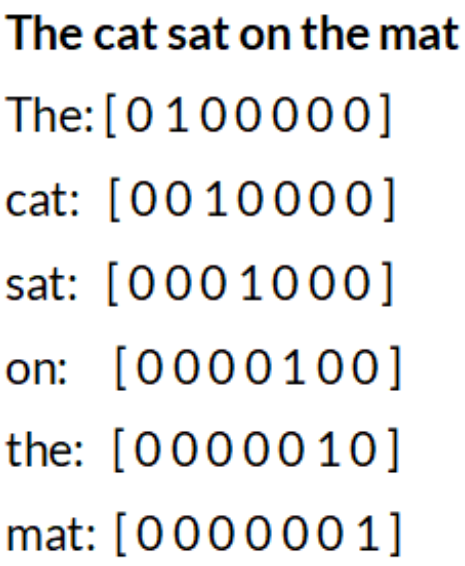
Chương 3

Mô hình Word2vec

Bài toán của chúng ta yêu cầu phải xử lý dữ liệu dưới 2 dạng là văn bản và hình ảnh. Với yêu cầu máy tính chỉ xử lý được dữ liệu dưới dạng số nên cả hai loại dữ liệu trên phải được đưa về dạng số. Dữ liệu hình ảnh có đầu vào là cường độ màu sắc và đã được mã hóa thành giá trị số trong khoảng [0, 255]. Còn dữ liệu văn bản có đầu vào chỉ là các chữ cái kết hợp với dấu câu. Do đó cần các phương pháp để lượng hóa được những đoạn văn bản này.

Cách truyền thống để thể hiện 1 từ là dùng one-hot-vector (hình 3.1).Tuy nhiên khi sử dụng phương pháp này thì vector từ được tao ra có số chiều rất lớn (bằng số lượng từ vựng). One -hot-vector cũng không thể hiện mối quan hệ giữa các từ tương đồng. Word2vec là phương pháp giải quyết cho vấn đề này. Có 2 mô hình Word2vec được áp dụng:

- Skip-gram
- Continuous Bag of Words (CBOW)



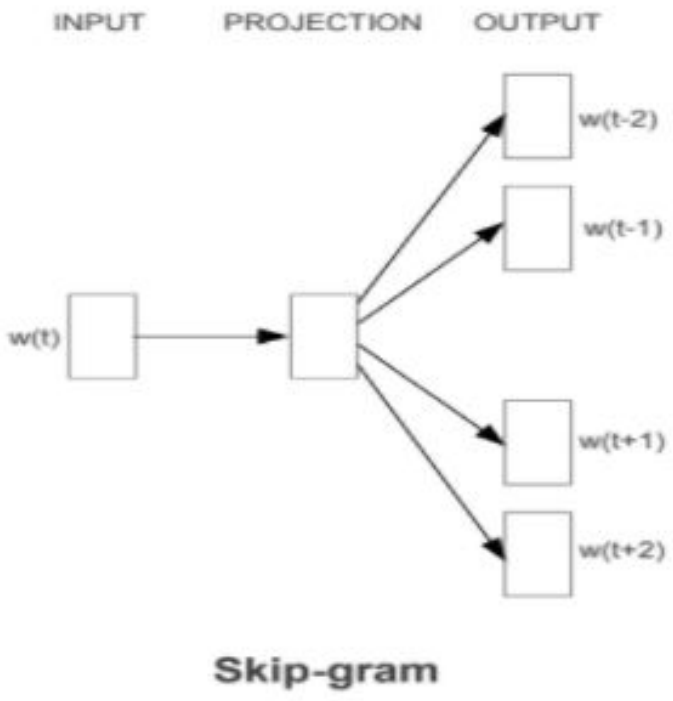
Hình 3.1: One hot vector cho các từ.

3.1 Skip-gram

Mô hình Skip-gram dự đoán các từ mục tiêu (target) dựa vào từ bối cảnh (context) trong chuỗi [1]. Điểm phân loại của các từ mục tiêu sẽ dựa vào mối quan hệ cú pháp với từ bối cảnh. Giả sử chúng ta có một câu văn như sau: "Trong các môn thể thao tôi thích bóng đá nhất". Để thu được một phép nhúng từ tốt hơn chúng ta sẽ lựa chọn ra ngẫu nhiên các từ làm bối cảnh (context). Dựa trên từ bối cảnh, các từ mục tiêu (target) sẽ được xác định nằm trong phạm vi xung quanh từ bối cảnh. Ví dụ ta sẽ chọn từ "thích" làm từ bối cảnh thì sẽ lần lượt thu được các từ mục tiêu sau:

Bối cảnh (context)	Mục tiêu (target)
thích	tôi
thích	bóng đá
thích	thể thao
thích	nhất

Các từ mục tiêu sẽ được giải thích tốt hơn nếu được học theo các từ bối cảnh. Do đó mô hình skip-grams tìm cách xây dựng một thuaatnj toán học có giám sát đầu vào là các từ bối cảnh và đầu ra là từ mục tiêu.
Mô hình Skip-gram sẽ có cấu trúc như hình dưới đây:



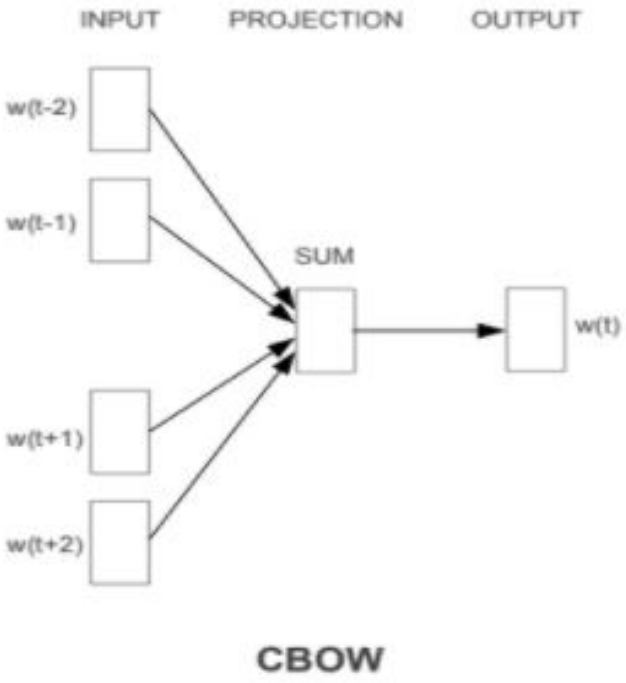
Hình 3.2: Kiến trúc mô hình skip-grams

Mô hình sẽ biểu diễn một từ bối cảnh dưới dạng one-hot véc tơ. Véctơ này sẽ trở thành đầu vào cho một mạng nơ ron có tầng ẩn gồm 300 units. Kết quả ở output layer là một hàm softmax tính xác xuất để các từ mục tiêu phân bố vào những từ trong vocabulary (10000 từ). Dựa trên quá trình feed forward và back propagation mô hình sẽ tìm ra tham số tối ưu để kết

quả dự báo từ mục tiêu là chuẩn xác nhất.
Mục đích chính của chúng ta làm tìm ra ma trận giữa input layer và hiddent layer. Các dòng của ma trận này chính là các vector nhúng đại diện cho mỗi từ bối cảnh. Kết quả dự báo mô hình mạng nơ ron càng chuẩn xác thì véc tơ nhúng sẽ càng thể hiện được mối liên hệ trên thực tế giữa từ bối cảnh và mục tiêu chuẩn xác.

3.2 CBOW

Mô hình skip-grams sẽ rất tốn chi phí để tính toán. Để hạn chế điều này thì mô hình CBOW (continueos backward model) được áp dụng. CBOW là một mô hình ngược lại của skip-gram. Nghĩa là thay vì dựa vào từ bối cảnh để dự đoán các từ mục tiêu thì CBOW dựa vào các từ mục tiêu để dự đoán từ bối cảnh. Kiến trúc mạng nơ ron của CBOW sẽ gồm 3 layers:



Hình 3.3: Kiến trúc mô hình CBOW

- Input layers: Là các từ bối cảnh xung quanh từ mục tiêu.
- Projection layer: Lấy trung bình véc tơ biểu diễn của toàn bộ các từ input để tạo ra một véc tơ đặc trưng.
- Output layer: Là một dense layers áp dụng hàm softmax để dự báo xác xuất của từ mục tiêu.

Mục đích chính của chúng ta làm tìm ra ma trận giữa input layer và hiddent layer. Các dòng của ma trận này chính là các vector nhúng

Chương 4

Xây dựng mô hình xử lý bài toán tạo ảnh từ văn bản

4.1 Phát biểu bài toán

Bài toán tạo ảnh từ đoạn văn bản có thể được phát biểu như sau:

Cho một văn bản $\varphi(t)$ mô tả nội dung ảnh đầu vào . Đoạn văn bản này sẽ được nhúng quá một model word vector để trở thành một vector $h \in R^T$ .Ta cần xây dựng một ánh xạ :

$$\mathcal{G} : R^T \times R^Z \to R^D \tag{4.1}$$

$$\widehat{x} = \mathcal{G}(h, z) \in R^D \tag{4.2}$$

Với T là số chiều của vector nhúng, Z là số chiều của vector nhiễu và D là số chiều của ảnh. Mục tiêu là sinh ra ảnh $\widehat{x}$ có được sự hợp lí, chân thực với mắt người và đồng thời cũng phù hợp với đoạn văn bản ở đầu vào. Việc xây dựng ánh xạ sẽ được dẫn đường bởi ánh xạ:

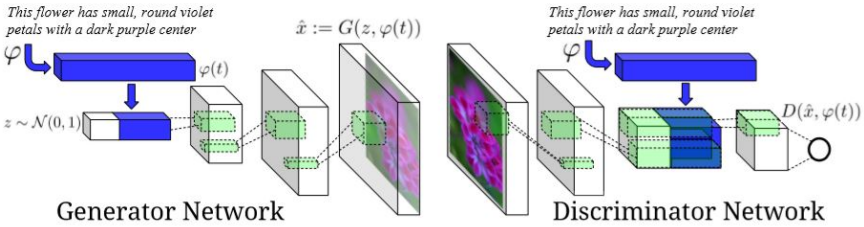
$$\mathcal{D} : R^D \times R^T \to 0, 1 \tag{4.3}$$

Trong mạng GAN được huấn luyện, bộ sinh G sẽ là ánh xạ $\mathcal{G}$ cần tìm. Còn ánh xạ $\mathcal{D}$ là bộ phân biệt D có nhiệm vụ phân biệt ảnh màu đưa vào là thật hay giả, hỗ trợ xây dựng ánh xạ. $\mathcal{G}$.

4.2 Kiến trúc của bài toán

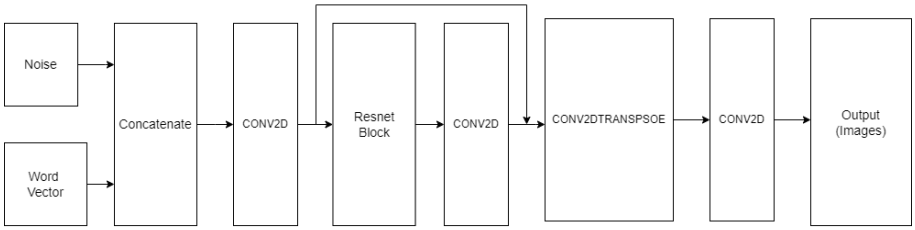
Kiến trúc được tham khảo từ Scott Reed [5]

Tưởng tự các mô hình GAN khác, mô hình của bài toán cũng gồm hai mạng là mạng sinh (Generator Network) và mạng phân biệt (Discriminator Network). Chúng ta sẽ đi tìm hiểu kỹ hơn kiến trúc của các mạng.



Hình 4.1: Kiến trúc mạng GAN của bài toán.

4.2.1 Mạng sinh (Generator Network)

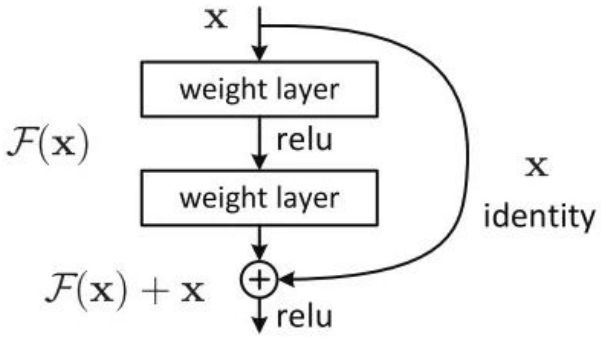


Hình 4.2: Kiến trúc mạng sinh của bài toán.

Mạng sinh nhận đầu vào là vector nhúng của đoạn văn bản là h (vector này có được sau khi nhúng đoạn văn bản $\varphi(t)$ qua mô hình word vector). Sau đó vector nhúng này sẽ được qua một lớp fully-connected và được reshape lại thành khối có kích thước 8x8x128. Tiếp theo lấy một vector nhiễu $z \in R^Z \sim \mathcal{N}(0, 1)$ và vector nhiễu z này cũng qua một lớp fully-connected và được reshape lại thành khối có kích thước 8x8x128. Sau đó 2 khối trên được concatenate với nhau thành khối có kích thước (8x8x256). Khối này sẽ qua một lớp tích chập với rồi đưa vào Resnet block.

Resnet block là khối gồm nhiều Residual Block , là mô hình mạng CNN sử dụng kết nối"tắt" đồng nhất để xuyên qua một hay nhiều lớp mạng. Một khối Residual Block có dạng như hình 4.3. Khối Resnet block trên sẽ bao gồm 4 khối Residual Block. Mỗi lớp tích chập trong khối có các thông số như sau: filters=64, kernel_size=3, strides=1, padding="same". Và sau mỗi lớp tích chập đều sử dụng một lớp BatchNormalization.
Sau khi qua khối Resnet Block thì dữ liệu sẽ đi qua một lớp tích chập nữa. Sau đó, dữ liệu sau lớp tích chập này sẽ được kết hợp với dữ liệu sau lớp tích chập ở trước khối Resnet Block như trong hình 4.2 để đưa vào khối giải chập (CONV2DTRANSPOSE).

Khối CONV2DTRANSPOSE bao gồm 4 lớp giải chập sử dụng hàm kích hoạt LeakyRelu. Dữ liệu sau khối này sẽ được đi qua một lớp tích chập cuối và cho ra ảnh có kích thước 64x64x3.



Hình 4.3: Residual Block.

4.2.2 Mạng phân biệt (Discriminator Network)

Mạng phân biệt nhận đầu vào thứ nhất là vector nhúng của đoạn văn bản là h (vector này có được sau khi nhúng đoạn văn bản $\varphi(t)$ qua mô hình word vector) . Sau đó vector nhúng này sẽ được qua một lớp fully-connected và được reshape lại thành khối có kích thước 64x64x3. Đầu vào thứ 2 là ảnh có kích thước 64x64x3 .Sau đó ảnh và khối vector nhúng trên được concatenate với nhau thành khối có kích thước (64x64x6). Khối này sẽ qua một 7 lớp tích chập. Sau khi qua 7 lớp tích chập , dữ liệu sẽ được flatten rồi qua 2 lớp fully-connected để cho ra kết quả cuối cùng 0,1. Với 0 tương ứng là ảnh giả và 1 tương ứng với ảnh thật.

4.3 Chu trình huấn luận và hàm mất mát

Mục tiêu của bài toán là xây dựng mô hình tạo ảnh chân thật và phù hợp với nội dung đoạn văn bản mô tả. Do đó bộ phân biệt không chỉ phân biệt giữa các cặp (ảnh giả, tiêu đề đúng), (ảnh thật, tiêu đề đúng) mà còn phải phân biệt thêm trường hợp (ảnh thật, tiêu đề sai). Do đó chu trình huấn luyện và hàm mất mát sẽ có dạng như sau:

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0,1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

Hình 4.4: Chu trình huấn luyện và hàm mất má

Chương 5

Triển khai mô hình

5.1 Dữ liệu

5.1.1 Dữ liệu ảnh

Dữ liệu là ảnh các loài chim được lấy từ tập dữ liệu CUB_200_2011 bao gồm 117888 tấm ảnh với 200 loại khác nhau (hình 5.1).

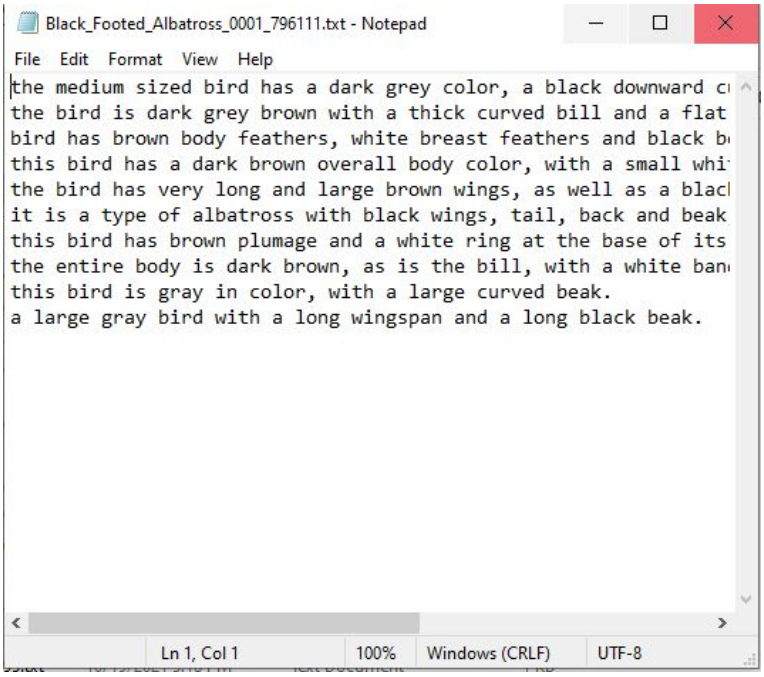
Hình 5.1: Dữ liệu từ tập CUB_200_2011 .

Chuẩn hoá dữ liệu đầu vào là bước tiền xử lí giúp tăng tốc độ hội tụ cho mô hình, giảm sự phụ thuộc của gradient vào tỉ lệ các tham số và một số lợi ích khác. Ảnh đầu vào của chúng ta cũng được chuẩn hóa để việc train mô hình được hiệu quả hơn.

5.1.2 Dữ liệu văn bản

Mỗi ảnh đều rất nhiều tiêu đề mô tả nội dung cho ảnh được viết bởi con người (hình 5.2).

Dữ liệu văn bản sau khi qua một số bước tiền xử lý sẽ được đưa vào bộ mã hóa wordvector **GoogleNews-vectors-negative300** để trở thành các vector nhúng từ .



Hình 5.2: Đoạn văn bản mô tả nội dung.

5.2 Huấn luyện mô hình

Các dữ liệu được xử lý và tải lên google drive. Mô hình sẽ được train trực tiếp trên Google Colab với sự trợ giúp của thư viện tensorflow.
Vì mô hình bộ sinh phức tạp và quan trọng hơn so với bộ phân biệt, vậy nên để cho quá trình huấn luyện GAN được ổn định hơn, cũng như tránh bộ phân biệt hội tụ sớm, ta sẽ tiền huấn luyện độc lập bộ sinh bằng qua 20 epoch với kích thước batch là 64. Mỗi epoch mất khoảng 2-3 phút khi huấn luyện trên Google Colab.

Tải các thông số của bộ sinh đã được tiền huấn luyện độc lập trước đó, đưa vào quá trình huấn luyện một mạng đối nghịch. Thuật toán tối ưu vẫn sẽ dùng là Adam [3] với tốc độ học $\alpha = 0.0000035$ và mô men $\beta_1 = 0.5$ .Mô hình được huấn luyện qua 600 epoch với kích thước batch là 64. Mỗi epoch mất khoảng 2–3 phút bằng việc sử dụng Google Colab để thực thi.

Chương 6

Kết quả

Ta sẽ thử nghiệm với 2 trường hợp là nội dung ảnh có trong dữ liệu train và nội dung ảnh tự tạo.

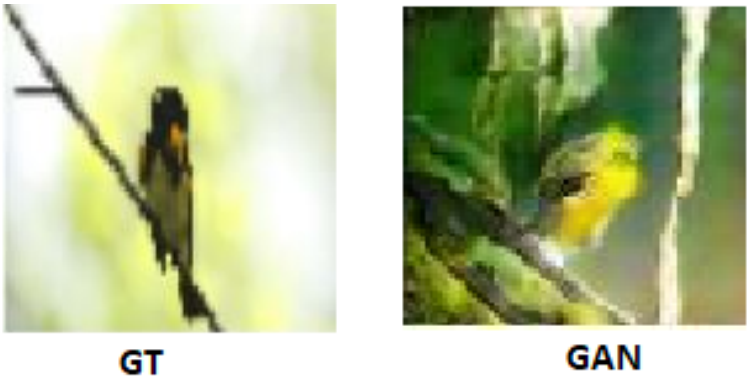
6.1 Thử nghiệm với nội dung mô tả trong dữ liệu train

- Nội dung 1 :This bird has a white belly, grey wings. (Tạm dịch: Một con chim có bụng trắng, cánh xám, chân và mỏ màu vàng.
  Hình 6.1 cho thấy với nội dung 1 thì ảnh được tạo bởi bộ sinh của GAN có bố cục và độ

Hình 6.1: Ảnh ứng với nội dung 1.

chân thực chưa cao, ảnh cũng chưa sát với nội dung và khác rất xa so với ảnh gốc. Chất lượng ảnh tạo ra thấp.

- Nội dung 2 :The small bird has a black head with yellow wing detail and a white body.(Tạm dịch: Con chim nhỏ có đầu đen với chi tiết cánh màu vàng và thân màu trắng.



Hình 6.2: Ảnh ứng với nội dung 2.

Hình 6.2 cho thấy với nội dung 2 thì ảnh được tạo bởi bộ sinh của GAN có bố cục và độ chân thực khá cao, ảnh cũng chưa sát với nội dung và khác rất xa so với ảnh gốc.Chất lượng ảnh tạo ra thấp.

- Nội dung 3: This bird is blue and white and has a very short beak.(Tạm dịch: Con chim này có màu trắng, xanh dương và có một cái mỏ rất ngắn.Chất lượng ảnh tạo ra trung bình.

Hình 6.3: Ảnh ứng với nội dung 3.

Hình 6.3 cho thấy với nội dung 3 thì ảnh được tạo bởi bộ sinh của GAN có bố cục và độ chân thực cao. Tuy nhiên không phù hợp với nội dung đưa ra.

- Nội dung 4: This small bird is brown with black symetrical highlights and a small beak. (Tạm dịch: Con chim nhỏ này có màu nâu với những điểm nổi bật đối xứng màu đen và một chiếc mỏ nhỏ.

Hình 6.4: Ảnh ứng với nội dung 4.

Hình 6.4 cho thấy với nội dung 4 thì ảnh được tạo bởi bộ sinh của GAN có bố cục và độ chân thực cao. Tuy nhiên ảnh chỉ phù hợp một phần với nội dung đưa ra.

- Nội dung 5: This is a small, yellow bird with black on the wingbars. (Tạm dịch: Đây là một con chim nhỏ, màu vàng với màu đen trên cánh quạt.

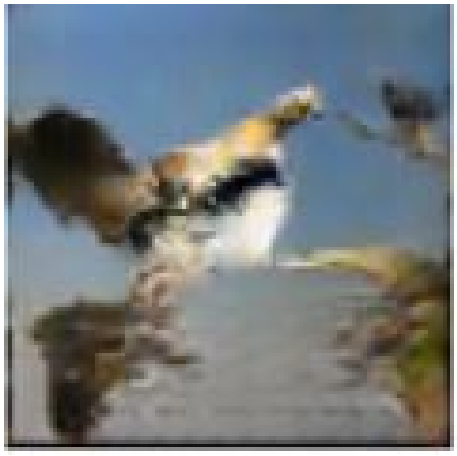
Hình 6.5: Ảnh ứng với nội dung 5.

Hình 6.5 cho thấy với nội dung 5 thì ảnh được tạo bởi bộ sinh của GAN có độ chân thực rất kém và không có ý nghĩa.

6.2 Thử nghiệm với nội dung tự tạo

Ta sẽ tự tạo một số nội dung và đưa vào mô hình sinh để tạo ảnh mới.

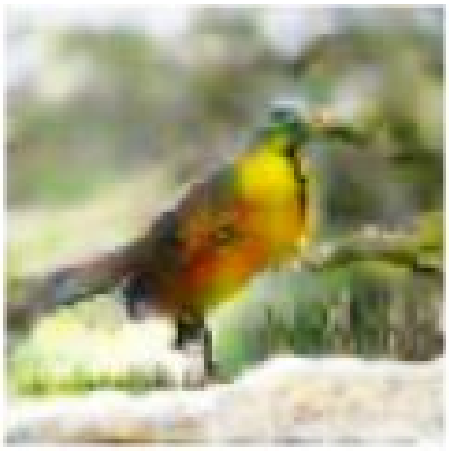
- Nội dung 1: This bird has white breast with brown feathers. (Tạm dịch: Con chim này có bộ ngực màu trắng với lông màu nâu).



Hình 6.6: Ảnh được tạo ứng với nội dung 1.

Ảnh 6.6 có bố cục và màu sắc phù hợp với đoạn văn. Tuy nhiên độ chân thực chưa cao và chất lượng ảnh còn thấp.

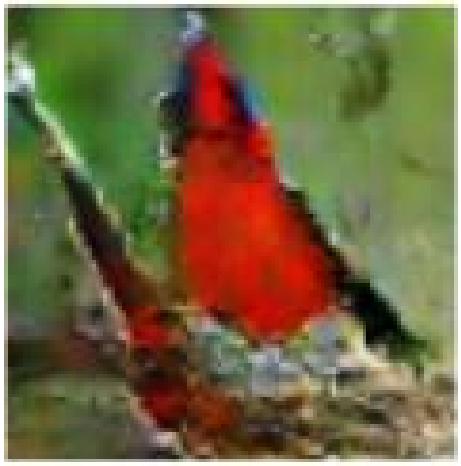
- Nội dung 2: This bird is completely yellow. (Tạm dịch: Con chim này hoàn toàn có màu vàng.).



Hình 6.7: Ảnh được tạo ứng với nội dung 2.

Ảnh 6.7 có bố cục và màu sắc phù hợp với đoạn văn.Độ chân thực và chất lượng ảnh cao.

- Nội dung 3: This bird is completely red. (Tạm dịch: Con chim này hoàn toàn có màu đỏ.).



Hình 6.8: Ảnh được tạo ứng với nội dung 3.

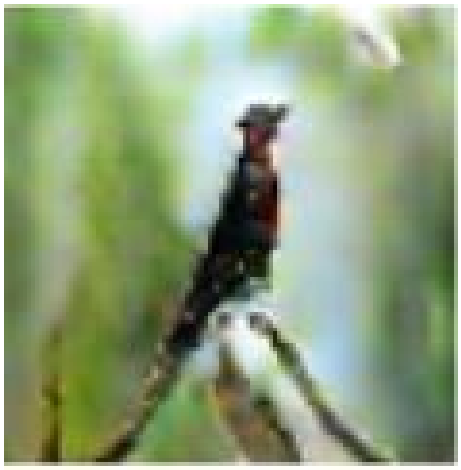
Ảnh 6.8 có bố cục và màu sắc phù hợp với đoạn văn.Độ chân thực và chất lượng ảnh cao.

- Nội dung 4: This bird is completely blue. (Tạm dịch: Con chim này hoàn toàn có màu xanh dương.).

Hình 6.9: Ảnh được tạo ứng với nội dung 4.

Ảnh 6.9 có bố cục và màu sắc phù hợp với đoạn văn.Độ chân thực và chất lượng ảnh cao.

- Nội dung 5: Orange bird with black wings and head feathers. (Tạm dịch: Con chim màu cam với đôi cánh và lông đầu màu đen.).



Hình 6.10: Ảnh được tạo ứng với nội dung 4.

Ảnh 6.10 có bố cục và độ chân thực khá cao. Tuy nhiên nội dung ảnh chỉ đúng một phần so với đoạn văn mô tả.

Chương 7

Tổng kết

7.1 Kết luận

- Ưu điểm: Xây dựng thành công mô hình tạo ảnh từ nội dung đoạn văn bản. Ảnh sinh ra từ đoạn văn có nội dung đơn giản có độ chân thực khá cao.
- Nhược điểm: Nếu nội dung đoạn văn bản phức tạp, ảnh tạo ra có thể sẽ bỏ lỡ một số chi tiết. Mô hình hoạt động chưa tốt khi hầu như các ảnh sinh ra đều có độ chân thực chưa cao, chất lượng kém.

7.2 Hướng phát triển

- Để khắc phục những hạn chế trên cần xây dựng mô hình tốt hơn. Có thể xây dựng mô hình word vector riêng cho bài toàn này. Mô hình chỉ hoạt động dựa trên mã hóa đoạn văn bản với mức độ là từng câu. Điều đó có thể làm mất một số đặc trưng trong câu nên cần phương pháp phân tích từng từ trong đoạn mô tả để không bỏ lỡ những chi tiết quan trong.
- Xây dựng mô hình triển khai trên tiếng Việt.

Tài liệu tham khảo

[1] Phạm Đình Khánh. "Mô hình Word2Vecs". In: *Khoa học dữ liệu - Khanh's blog* (2019). URL: https://phamdinhkhanh.github.io/2019/04/29/ModelWord2Vec.html.

[2] Phạm Đình Khánh. "Model GAN". In: *Khoa học dữ liệu - Khanh's blog* (2020). URL: https://phamdinhkhanh.github.io/2020/07/13/GAN.html.

[3] Diederik P. Kingma and Jimmy Lei Ba. *ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION*. 2017. arXiv: 1412.6980 [cs.LG].

[4] Manisha Padala, Debojit Das, and Sujit Gujar. *Effect of Input Noise Dimension in GANs*. 2020. arXiv: 2004.06882 [cs.LG].

[5] Scott Reed. *Generative Adversarial Text to Image Synthesis*. 2016. arXiv: 1605.05396 [cs.LG].

[6] tejaskhot. "Reply: Optimization metric in Generative Adversarial Networks". In: *https://stats.stackexchange.com/* (2017). URL: https://bit.ly/3qvtnfl.

[7] Wikipedia. "Cross entropy". In: *https://en.wikipedia.org/wiki/* (2021). URL: https://en.wikipedia.org/wiki/Cross_entropy.

[8] Wikipedia. "Vanishing gradient problem". In: *https://en.wikipedia.org/wiki/* (2021). URL: https://en.wikipedia.org/wiki/Vanishing_gradient_problem.