DOI 10.31891/2307-5732-2023-329-6-181-182
УДК 621

КОПАЧ БОГДАН
Національний університет «Львівська політехніка»
https://orcid.org/0009-0002-5158-589X
e-mail: bohdan.v.kopach@lpnu.ua

ВПЛИВ МОРФОЛОГІЇ ШАРІВ ТРАНСФОРМАЦІЇ ВЕКТОРІВ ТЕКСТУ ТА ЗОБРАЖЕННЯ НА ТОЧНІСТЬ CLIP МОДЕЛІ

Пошук шляхів для знаходження взаємозв'язків між зображеннями та текстом є складним завданням, вирішення якого ускладнюється великою кількість можливих варіантів, форм, представлень однакових об'єктів як на зображеннях, так і за допомогою текстового опису. Із моменту релізу CLIP моделі у 2021 році ця сфера активно розвивається, на її основі почали формуватися моделі, які активно використовуються для створення зображень за текстовим описом, доповнюють та описують зображення тощо. Актуальність дослідження полягає у вивченні та вдосконаленні методів аналізу взаємов'язків між текстовими та візуальними даними в передових моделях штучного інтелекту, які використовують декілька нейронних мереж, зокрема таких, як CLIP. Це дозволяє покращити точність та ефективність обробки інформації, що має велике значення в багатьох сферах, наприклад, завданнях комп'ютерного зору та автоматичного опрацювання природної мови. Головна мета цієї статті – дослідження впливу зміни структури шарів трансформації CLIP моделі, що відповідають за зміну довжини векторів тексту та зображення, на її точність. На етапі проведення експериментів використовувалися кодувальники зображень на основі ResNet-50 та ViT-B/32, кодувальник тексту BERT та різні комбінації й типи прихованих шарів нейронної мережі. Отримані результати показують, що застосування декількох лінійних шарів із шаром нормалізації та поступове зменшення довжини векторів даних може покращити точність CLIP моделі на 10-15% в залежності від функції втрат, що використовується для навчання, та кодувальників зображень. Визначено, що різке зменшення довжини векторів, які репрезентують текстові та візуальні дані, або використання занадто великої кількості нейронних шарів для їх опрацювання може негативно впливати на точність CLIP моделі. Запропоновані архітектурні рішення дозволяються покращити здатність моделі знаходити взаємозв'язки між зображеннями та текстом.

Ключові слова: нейронні мережі, CLIP, опис зображення, векторні перетворення.

KOPACH BOHDAN
Lviv Polytechnic National University

INFLUENCE OF THE MORPHOLOGY OF TEXT AND IMAGE VECTOR TRANSFORMATION LAYERS ON THE ACCURACY OF THE CLIP MODEL

Searching for ways to establish relationships between images and text is complex, due to the vast array of variations, forms, and representations of identical objects in both mediums. Since the CLIP model's introduction in 2021, the field has seen rapid growth, leading to the development of new models based on CLIP. These are extensively used for generating images from text, image inpainting, and image description. The significance of this research lies in enhancing methods for analyzing the interplay between text and visual data in advanced AI models, like CLIP, which employ multiple neural networks. This enhancement is crucial for improving accuracy and efficiency in processing information, which is particularly important in computer vision and natural language processing. The primary aim of this study is to explore how modifications in the transformation layers of the CLIP model, which adjust the lengths of text and image vectors, affect its accuracy. The experiments utilized image encoders based on ResNet-50 and ViT-B/32, the text encoder BERT, and various combinations and types of neural network's hidden layers. The results demonstrate that using multiple linear layers with a normalization layer and progressively shortening the data vectors can enhance the CLIP model's accuracy by 10-15%, varying with the loss function and image encoders used in training. However, significantly reducing the vector lengths for textual and visual data, or employing too many neural layers for processing, can detrimentally affect the model's accuracy. The architectural solutions proposed in the research are tailored to address these challenges. They focus on optimizing the morphology of transformation layers and carefully adjusting the size of the vectors to ensure that the model retains enough information for accurate analysis while not being burdened by unnecessary data or complexity. The study not only contributes to the ongoing development of more accurate and efficient AI models for handling complex text and image relationships but also provides insights into the importance of balance and precision in AI architecture design.

Keywords: neural networks, CLIP, image description, vector transformations.

Постановка проблеми

Нейронні мережі, які можуть знаходити залежності між зображення та текстом, будувати деталізований опис зображення, який складається не лише із набору заздалегідь визначеного набору класів, а й з семантично зв'язного тексту із врахуванням стану об'єктів, що знаходяться на зображенні, завжди були об'єктом досліджень науковців. Активний розвиток моделей для розпізнавання зображень, поява великих мовних моделей BERT [1] та GPT-2 [2] лише пришвидшили розвиток інструментарію для встановлення семантичних зв'язків між текстом та зображенням. Нейронні мережі почали тренувати, отримуючи векторне представлення тексту та семантично зв'язуючи його із векторним представленням зображення, навчаючи модель на парах текст-зображення – ConVirt [3] або навчаючи передбачати замасковану частину тексту за зображеннями – ICMLM [4]. Складність навчання подібних моделей та придатність для вирішення лише певного типу завдань значно обмежує сферу їхнього використання.

Поява CLIP [5] стала важливим етапом розвитку універсальних багатомодельних нейронних мереж, які можна використовувати в широкому спектрі завдань, зокрема класифікації зображень, отримання текстового опису зображення або пошуку зображень за текстом. CLIP використовує контрастивне навчання,

намагаючись поєднати пари текст-зображення, навчаючи паралельно кодувальник тексту та зображення. Для навчання моделі можна задіювати заздалегідь натреновані кодувальники, а запропонована архітектура достатньо гнучка, щоб використовувати різні функції втрат, оптимізатори тощо. У наш час з'являється багато модифікацій CLIP моделі, а різні її види адаптуються до вирішення специфічних завдань, таких як: розпізнавання зображень [6], пошук контенту із образливим вмістом [7] або опис фрагментів відео [8]. Разом із цим залишається багато питань до структури CLIP моделі та можливостей її оптимізації.

Мета роботи – дослідження впливу структури шарів трансформації векторів тексту та зображення на точність CLIP моделі, зокрема:

- експериментальному визначенні оптимальної структури нейронної мережі для зменшення розміру векторів ознак, отриманих від кодувальників тексту та зображень, для збільшення точності передбачень моделі та правильному визначенні зв'язків текст-зображення, зображення-текст;
- дослідженні впливу зміни функції втрат, що використовується на етапі навчання моделі, на її точність залежно від структури нейронної мережі, зокрема шарів трасформації векторних представлень.

Під час проведення експериментів для кодування тексту використовувалася модель BERT. Її застосування як кодувальника тексту зумовлене універсальністю, точністю, простотою навчання та можливістю порівняння семантичної спорідненості тексту за допомогою векторного представлення речень.

Описані в цій статті методи та висновки можуть використовуватися для покращення точності будь-якої CLIP моделі, де текстовим кодувальником є BERT, та не обмежені специфічною галуззю науки або техніки. Попри це важливо розуміти, що, методи та підходи до налаштування параметрів можуть відрізнятися залежно від конкретних цілей та даних, що застосовуються під час навчання моделі.

Аналіз останніх досліджень

Зміни в морфології CLIP моделі зазвичай диктуються вимогами та обмеженнями, які на неї накладаються. Більшість наукових досліджень CLIP сконцентровані на кодувальниках тексту та зображення. Вони є критичною частиною структури CLIP, оскільки відіграють ключову роль у перетворенні вхідних даних на високорівневі, структуровані векторні представлення.

У статті [5] було вперше описано CLIP модель, у якій для того, щоб перетворити вектори тексту та зображення у вектори однакової довжини, використовувався всього лиш один прихований лінійний шар. Основним недоліком такого підходу є те, що при значній зміні довжини будь-якого із векторів, що сформовані за допомогою кодувальників, частина даних може бути втрачена.

Навчання моделей, які використовують декілька нейронних мереж, потребує великої кількості обчислювальних ресурсів, тому збільшення кількості прихованих шарів, які опрацьовують вектори тексту та зображення, може призвести до того, що час навчання та опрацювання вхідних даних збільшується до критичної межі, яка робить недоцільним використання CLIP моделі в системах, де час відповіді є визначальним параметром. У статті [9] запропоновано спосіб вирішення цієї проблеми. Модифікована CLIP модель об'єднує вектори ознак тексту та зображень і використовує звичайний лінійний класифікатор для отримання відповіді на запитання до зображення. Недоліком такого підходу є те, що така структура моделі накладає певні обмеження на способи її використання та її масштабування. Така CLIP модель може використовуватися лише для вирішення завдання класифікації і не вивчає двосторонніх зв'язків між текстом та зображенням.

У роботі [10] описано навчання CLIP моделі із використанням додаткового кодувальника, який навчається за допомогою сценічного графу (структурного представлення зображення, яке описує об'єкти, їхні взаємозв'язки та атрибути). При опрацюванні запитань до зображення або при генерації опису модель, що використовує сценічний граф, розглядає зображення не лише як набір пікселів, а надає більш деталізовані та точні відповіді, зосереджуючись на конкретних взаємовідношеннях між об'єктами. Автоматичне створення точних сценічних графів вимагає додаткових моделей розпізнавання об'єктів, класифікації атрибутів та визначення взаємозв'язків. Це значно ускладнює процес навчання моделі. Але інтеграція сценічних графів в CLIP модель значно збільшує обчислювальні вимоги, особливо при опрацюванні зображень із великою кількістю різноманітних об'єктів. Також проблемою є те, що сценічний граф, створений на основі застарілих даних, не завжди може опрацювати нові об'єкти або рідкісні взаємозв'язки між ними, які виникають лише за певних умов.

На відміну від підходу, описаного у [10], де основною модифікацією CLIP моделі є використання додаткового кодувальника для аналізу структурних особливостей зображення, у досліджені [11] під час навчання моделі для кожного зображення створюється граф подій, центральним вузлом якого є конкретна дія. Вона поєднується із іншими вузлами (сутностями), використовуючи ролі. За допомогою графу формується набір позитивних та негативних описів зображення, які потім застосовуються для навчання моделі. Основним недоліком такого підходу є складність формування деталізованого графу, що може значно сповільнити процес навчання CLIP моделі. Також слід зауважити, що для зведення векторів тексту та зображення до однакового розміру, їхня довжина обмежується до однакового значення. Це теж негативно впливає на результати, адже частина інформації може бути втрачена.

У статті [12] запропоновано стратегію дистиляції (передачі) знань від однієї CLIP моделі до іншої. Під час дистиляції модель-«учитель» передає свої знання «учню». Найчастіше метою такого процесу є оптимізація процесу навчання або створення моделі, яка є меншою за «вчителя», але при цьому зберігає

високий рівень точності. Для навчання CLIP моделей зазвичай використовують дистиляцію ознак тексту та зображень [12, 13]. Однак слід враховувати, що «учень» може перейняти всі помилки «вчителя» та інтегрувати їх у свої прогнози. Під час передачі знань важливо враховувати архітектурні відмінності обох моделей, оскільки у випадку, якщо модель-«учитель» використовує відмінні від «учня» типи нейронних мереж для видобування ознак тексту та зображень, може виникнути потреба у структурних змінах CLIP моделі.

Формулювання цілі статті

Дослідження взаємозв'язків зображення-текст було одним із основних завдань комп'ютерного зору упродовж років. Реліз Open AI CLIP моделі став важливим кроком на шляху до вирішення цього завдання. Для навчання моделі використовуються пари текст-зображення, які передаються в кодувальники, що перетворюють дані у векторне представлення.

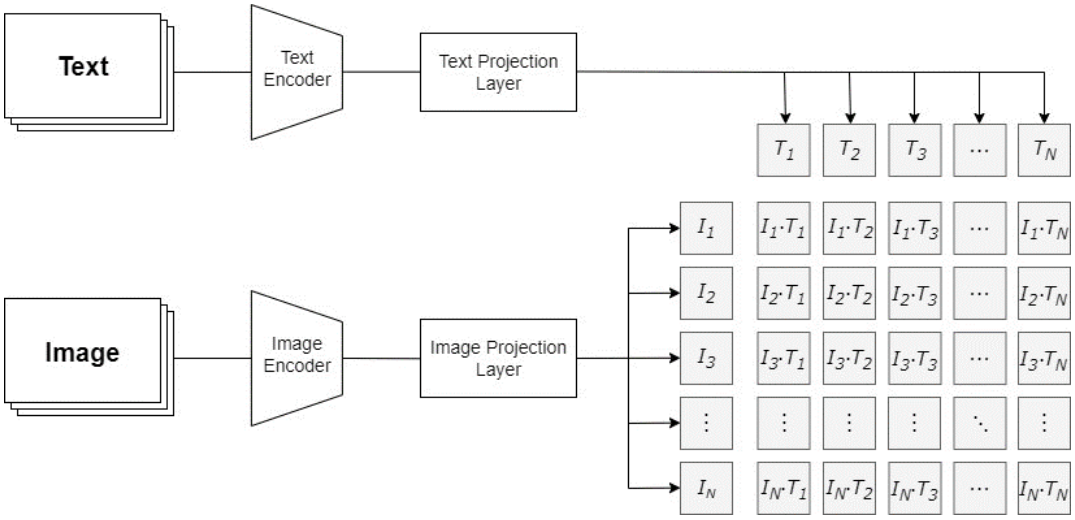

Рис. 1. Схема тренування CLIP моделі; Image Encoder – кодувальник зображення; Text Encoder – кодувальник тексту; Text Projection Layer – шар трансформації векторного представлення тексту; Image Projection Layer – шар трансформації векторного представлення зображення

Для того, щоб перетворити зображення та текст у векторну форму, яка може бути використана для обчислень, використовуються кодувальники. Нехай дано пари терм (T) – зображення (I), тоді процес їхнього перетворення у вектор можна представити наступною формулою:

$$\overline{v_{I_i}} = f_I(I_i) \tag{1}$$

$$\overline{v_{T_i}} = f_T(T_i), \tag{2}$$

де i – порядковий номер пари, $i \in \{1,2,..,n\}$, n – кількість пар;

$\overline{v_{I_i}}$ – елемент *i*-го векторного представлення зображення;

$\overline{v_{T_i}}$ – елемент *i*-го векторного представлення текстового опису;

f_I – кодувальник зображення;

f_T – кодувальник тексту;

Тоді завданням навчання CLIP моделі є оптимізація параметрів функції S, яка може бути використана для знаходження відповідності між парами зображення-текст:

$$sim(\overline{v_{I_i}}, \overline{v_{T_i}}) = S(\overline{v_{I_i}}, \overline{v_{T_i}}) \tag{3}$$

Припустимо що у нас є текстовий опис T', який не відповідає зображенню, із цього випливає:

$$sim(\overline{v_{I_i}}, \overline{v_{T_i}}) >> S(\overline{v_{I_i}}, \overline{v_{T_i'}}) \tag{4}$$

Однією із ключових проблем тренування CLIP моделі є те, що вектори, які створили кодувальники, зазвичай мають різну розмірність. Вища розмірність може містити більш детальне представлення початкової інформації, але слід врахувати, що чим більший вектор, тим більше необхідно обчислювальних ресурсів для роботи із ним. У CLIP шаром, який відповідає за приведення векторів до одного розміру, є шар трансформації (проєкції).

Процес отримання векторного представлення тексту та зображення дещо відрізняється. Для отримання векторного представлення тексту за допомогою GPT або BERT необхідно перетворити текст у токени (частини слів) та передати їх у моделі, виходом є вектор фіксованого розміру. Для отримання векторного представлення зображення зазвичай використовують механізм видобування ознак. Він полягає в тому, що із заздалегідь натренованої нейронної мережі отримують числове представлення зображення,

зазвичай використовують один із шарів нейронної мережі перед шаром класифікації або комбінацію декількох шарів. У табл. 1 подано перелік нейронних мереж та типові розміри векторів зображень або тексту.

Таблиця 1

Розмір векторних представлень моделей, що використовуються для кодування тексту або зображень

Модель	Тип даних	Довжина вектору	Типовий шар, що використовується для видобування ознак
BERT-base [1]	Текст	768	Останній прихований шар трансформера
GPT-base [2]	Текст	768	Останній прихований шар трансформера для кожного токена, зазвичай використовують середнє значення
VGG-19 [14]	Зображення	4096	Повноз'єднаний шар перед класифікаційним шаром
ResNet-50 [15]	Зображення	2048	Повноз'єднаний шар перед класифікаційним шаром
ViT-B/32 [16]	Зображення	768	Останній прихований шар трансформера

Різний розмір векторів ускладнює їх опрацювання. Обчислення, що проводяться над невеликими векторами, потребують менше обчислювальних ресурсів та виконуються швидше. Однак при значному зменшенні розміру вектора існує ризик втрати критично важливої інформації, що може негативно вплинути на точність моделі та якість отриманих результатів. З іншого боку, використання занадто великих векторів може призвести до зайвих витрат ресурсів, затримок у відгуках системи та зменшити ефективність моделі. Саме тому важливо шукати такі способи перетворення векторних представлень, які дозволили б мінімізувати вплив векторних перетворень на точність CLIP моделі, забезпечуючи зменшення розмірності так, щоб зберігати найважливіші аспекти даних.

Ціллю статті є пошук шляхів вирішення проблеми трансформації векторних представлень зображення та тексту CLIP моделі за допомогою нейронної мережі. Ідея полягає в тому, щоб навчити нейронну мережу перетворювати вектори різного розміру до однакової довжини, при цьому максимально зберігати корисну інформацію про текст та зображення.

Виклад основного матеріалу

CLIP модель працює із двома окремими векторами, кожен із яких містить інформацію про інший тип даних, саме тому важливо використовувати архітектуру нейронних мереж, яка може адаптуватися до різних типів даних. Це дозволить створити два різних шари перетворення векторів, кожен із яких навчатиметься окремо і може бути налаштований так, щоб максимально адаптуватися до свого типу даних, це покращуватиме загальну продуктивність системи. У випадках, коли для одного типу даних доступний менший набір даних, окремий шар перетворення може фокусуватися на цьому конкретному типі, не впливаючи на загальну продуктивність моделі.

Нейронна мережа, яка використовується для зменшення векторів у CLIP, повинна мати можливість адаптуватися до числових векторів будь-якого розміру та здійснювати їх перетворення із збереженням початкової інформації. Нехай $\overline{V_I}$ та $\overline{V_T}$ вектори із довжиною L_i та L_T, тоді завданням нейронної мережі є знаходження структури, яка дозволила би зменшувати розмірність до розміру L_S, при чому $L_S \le \min(L_I, L_T)$.

Плавне зменшення розміру вектора із невеликим кроком дозволяє ефективніше вивчати ієрархічні особливості даних та робить мережу стійкішою до перенавчання. Раптове зменшення може призвести до втрати інформації та зменшує стабільність навчання. Саме тому для трансформації векторів використовуватимемо декілька блоків, кожний із яких буде зменшувати розмірність вектору на крок Δ. Тоді кількість шарів N для вектору довжиною L можна розрахувати за наступною формулою:

$$N = \frac{L - L_S}{\Delta} \tag{5}$$

У цій статті було досліджено декілька типів нейронних блоків, які можуть використовуватися як основа для шарів зменшення довжини векторів CLIP моделі (рис. 2). Кожен із екземплярів має свої особливості й наукову цінність для проведення емпіричних досліджень.

Важливо зазначити, що велика кількість блоків може призвести до втрати важливої інформації та збільшити час навчання в рази, тоді як занадто мала кількість блоків не забезпечуватиме достатнього зменшення розмірності. Тому оптимальний вибір значення T є важливим для досягнення бажаного балансу між збереженням інформації та ефективністю обчислень.

CLIP модель використовує контрастивну функцію втрат, яка гарантує те, що подібні зображення та їх текстовий опис наближаються одне до одного у векторному просторі, тоді як невідповідні пари текстів та зображень віддаляються [5]. Найбільш широковживаною функцією втрат для навчання CLIP моделі є функція втрат на основі перехресної ентропії:

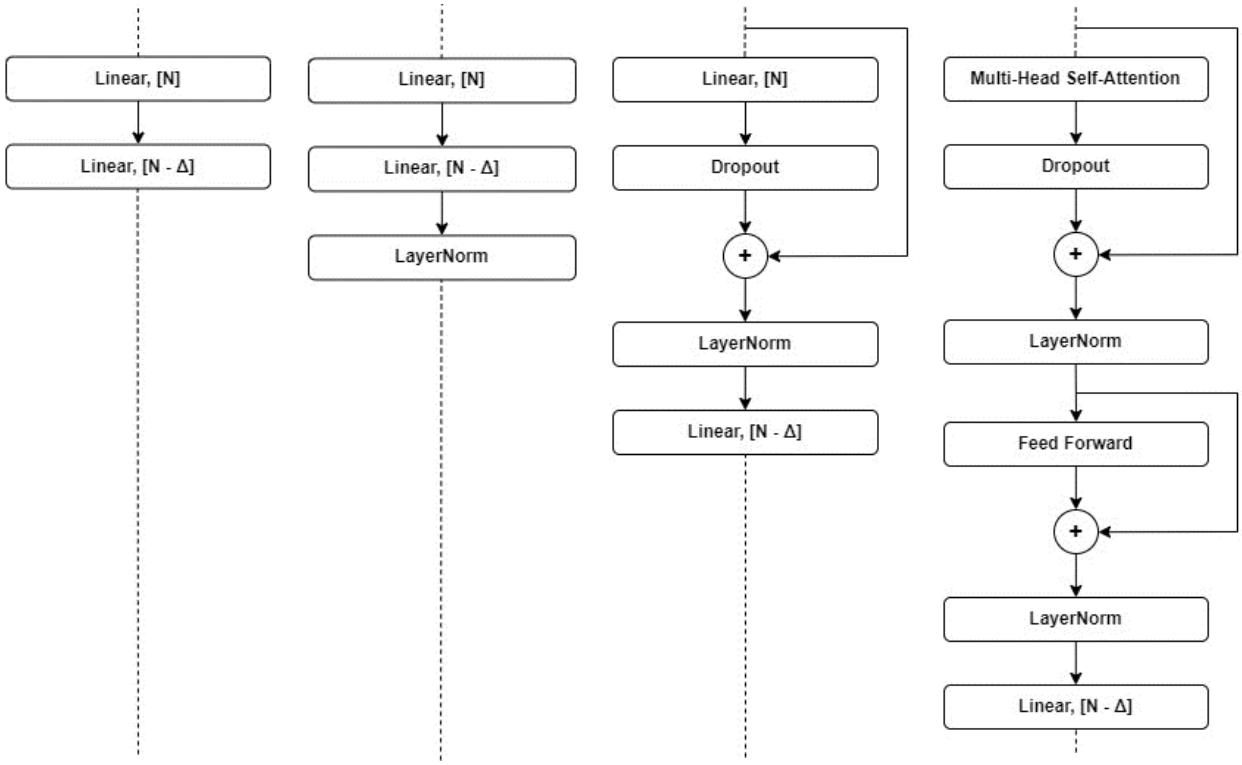

Рис. 2. Блоки нейронних мереж, що використовувалися для зменшення розміру векторів CLIP мережі: а – базовий приклад, який складається лише з лінійних прихованих шарів, б – нейронний блок містить додатковий шар нормалізації [17], в – мережа із пропущеним зв'язком, г – кодувальник із статті [16]; типи шарів: Linear – лінійні приховані шар, LayerNorm – шар нормалізації [17], Multi-Head Self-Attention – блок уваги[16], Feed Forward - декілька лінійних шарів [16], Dropout – шар регуляризації [18]

$$H(P,Q) = -\sum_{i=1}^{N} P(x)\log Q(x) \tag{6}$$

де $P(x)$ – розподіл відповідних пар;

$Q(x)$ – розподіл ймовірностей прогнозів моделі;

N – кількість елементів в одній ітерації навчання.

Нехай подібність між зображенням і текстом обчислюється за формулою:

$$S_{ij} = \overline{v_i}\overline{t_j}\,, \tag{7}$$

де $\overline{v_i}$ – векторне представлення i-го зображення;

$\overline{t_j}$ – векторне представлення j-го тексту.

Тоді значення втрат визначатиметься за формулою:

$$L_{ij} = -\log\frac{e^{S_{ij}}}{\sum_{i=1}^{N} e^{S_{ik}}} \tag{8}$$

Окрім функції перехресної ентропії, також важливо досліджувати вплив інших функцій втрат на точність моделі, оскільки вони є важливим індикатором динаміки навчання та дозволять встановити, як зміна шарів нейронної мережі, які відповідають за перетворення векторів, впливає на загальну ефективність моделі.

Однією із найбільш популярних альтернатив функції втрат перехресної ентропії є функція NT-Xent [19]. Основна ідея NT-Xent полягає в тому, щоб максимізувати подібність між позитивними парами, нормалізуючи вектори під час обчислення схожості (використовуючи косинусну подібність) та масштабуючи їх за допомогою параметру температури. Схожість між векторним представленням зображення $\overline{v_i}$ та тексту $\overline{t_j}$ обчислюватиметься за допомогою формули:

$$S_{ij} = \frac{\overline{v_i}\overline{t_j}}{\left\|\overline{v_i}\right\|\left\|\overline{t_j}\right\|} \tag{9}$$

Поділивши на «температурний» параметр τ, отримуємо:

$$S_{ij}^{'} = \frac{S_{ij}}{\tau} \tag{10}$$

Параметр τ контролює концентрацію оцінок. Чим нижче значення τ, тим більше пара зображення-текст віддалятиметься від усіх інших прикладів, які знаходяться в одному пакеті під час навчання моделі. Функція втрат NT-Xent для CLIP моделі визначатиметься за формулою [19]:

$$L_{ij} = -\log \frac{e^{S_{ij}^{'}}}{\sum_{k=1(i\neq k)}^{2N} e^{S_{ik}^{'}}} \tag{11}$$

Щоб отримати глибше розуміння того, як різні функції втрат впливають на точність CLIP моделі, зокрема у випадках, коли негативним прикладом є текстовий опис, який максимально відрізняється від зображення, використаємо функцію втрат триплетів [20]:

$$L_{ij} = \max\left(\frac{\overline{v_i}\,\overline{t_j}}{\left\|\overline{v_i}\right\|\left\|\overline{t_j}\right\|} - \frac{\overline{v_i}\,\overline{t_n}}{\left\|\overline{v_i}\right\|\left\|\overline{t_n}\right\|} + \alpha, 0 \right), \tag{12}$$

де $\overline{t_n}$ – векторне представлення текстового опис, який максимально відрізняється від очікуваного текстового опису i-го зображення.

CLIP розроблено для розуміння як візуальних, так і текстових даних. Функція втрат відіграє ключову роль у визначенні того, як ці два типи даних поєднуються, тому вибір правильної втрати може значно вплинути на продуктивність моделі. Враховуючи складність таких моделей, як CLIP, забезпечення стабільного навчання є критично важливим, оскільки правильно підібрана функція втрат гарантуватиме, що семантично схожі зображення або текстові фрагменти будуть ближче у векторному просторі.

Для попереднього навчання кодувальників зображень (ResNet-50 та ViT-B/32) використовувався набір даних ImageNet-1k [21], який містить близько одного мільйона фотографій, що належать до 1000 різних категорій, а для навчання CLIP моделі – публічні набори даних Flickr8k [22] та його розширення Flickr30k [23]. Вони часто застосовуються для навчання моделей, що спеціалізуються у розпізнаванні або описі зображень. Сумарно набори даних складаються із майже 40 тисяч ілюстрацій, кожна з яких містить 5 унікальних описів, які можуть мати різну довжину та стилістичне забарвлення.

Кожна із картинок проходила цілий ряд типових перетворень для уніфікації процесу навчання, зокрема зображення зменшувалися до розміру 256x256 пікселів, центральна частина зменшеного зображення обмежувалася до розміру 224x224 з метою концентрації на найважливішій частині зображення**.**

Навчання моделей здійснювалося за допомогою фреймворку PyTorch із використанням CUDA обчислень, а для оптимізації параметрів навчання застосовувався оптимізатор Адама [24] із кроком навчання 1e-5. Із метою перевірки, як зміна шарів перетворення векторів впливає на CLIP моделі, під час навчання використовувалися такі функції втрат: NT-Xent, перехресна ентропія та функція втрат триплетів у поєднанні із різними кодувальниками зображень та кодувальником тексту BERT.

Обчислення попарної косинусної подібності (13) між текстовими векторами, які CLIP вважає найкращими відповідниками для зображення та очікуваними текстовими описами, дозволяє кількісно виміряти ефективність моделі, надаючи об'єктивний критерій для оцінки її здатності правильно співвідносити візуальний та текстовий контент. Замість суб'єктивної оцінки відповідності тексту та зображення отримуємо числове значення, яке може бути використане для порівняння точності моделей.

$$S = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{N} \frac{\overline{E_i}\,\overline{A_k}}{\left\|\overline{E_i}\right\|\left\|\overline{A_k}\right\|}, \tag{13}$$

Таблиця 2

Результати тестування CLIP моделі на основі кодувальників ResNet-50 та BERT

Тип шару трансформації та функції втрат	Значення косинусної подібності в залежності від кількості нейромережевих блоків N			
	N=2	N=4	N=6	N=8
2а+CSE	0.44	0.48	0.48	0.46
2а+NtXent	0.46	0.49	0.52	0.51
2а+Triplet	0.36	0.36	0.37	0.34
2б+CSE	0.52	0.55	0.58	0.55
2б+NtXent	0.55	0.62	0.61	0.57
2б+Triplet	0.41	0.41	0.40	0.37
2в+CSE	0.45	0.52	0.53	0.53
2в+NtXent	0.45	0.53	0.52	0.46
2в+Triplet	0.42	0.41	0.43	0.38
2г+CSE	0.43	0.43	0.42	0.41
2г+NtXent	0.42	0.40	0.40	0.35
2г+Triplet	0.39	0.37	0.36	0.33

де N – кількість очікуваних текстових фрагментів, які описують зображення;

E – векторне представлення одного із очікуваних текстових описів;

A – векторне представлення текстового опису, визначеного за допомогою CLIP моделі, який найближчий до зображення у векторному просторі.

У табл. 2 наведено результати тестування CLIP моделі, що використовує кодувальник зображення ResNet-50 із різними типами блоків трансформації векторів (рис. 2) та функціями втрат. Найбільш ефективною виявилися функція втрат NT-Xent у поєднанні із блоками для перетворень із рис. 2б та 2в, що використовують шари нормалізації або зв'язки із пропущеннями.

У табл. 3 наведено результати тестування CLIP моделі, що використовує код увальник зображення ViT-B/32. Найбільш ефективною виявилися функція втрат перехресної ентропії у поєднанні із блоками для перетворень із рис. 2б та 2в.

Таблиця 3

Результати тестування CLIP моделі на основі кодувальників ViT-B/32 та BERT

Тип шару трансформації та функції втрат	Значення косинусної подібності в залежності від кількості нейромережевих блоків N			
	N=2	N=4	N=6	N=8
2а+CSE	0.48	0.51	0.47	0.46
2а+NtXent	0.48	0.50	0.48	0.45
2а+Triplet	0.41	0.40	0.42	0.39
2б+CSE	0.57	0.66	0.62	0.55
2б+NtXent	0.54	0.59	0.61	0.57
2б+Triplet	0.43	0.44	0.46	0.43
2в+CSE	0.56	0.61	0.58	0.57
2в+NtXent	0.52	0.55	0.58	0.52
2в+Triplet	0.44	0.45	0.41	0.42
2г+CSE	0.40	0.47	0.49	0.41
2г+NtXent	0.42	0.47	0.48	0.43
2г+Triplet	0.40	0.38	0.41	0.40

Експерименти проводилися за однакових умов, оскільки забезпечення стабільності процесу навчання було критично важливим для отримання об'єктивних результатів. З метою забезпечення порівнянності результатів, налаштування оптимізаторів та функцій втрат залишалися незмінними упродовж усіх експериментів. Такий підхід допомагає уникнути можливих спотворень на етапі тестування моделі.

Висновки

У цій роботі було проаналізовано, як використання різних типів нейронних мереж для зменшення довжини векторів впливають на точність CLIP моделі. Було спроєктовано декілька типів нейронних мереж та проведено експериментальне дослідження з використанням різних кодувальників та параметрів навчання моделі. Аналізуючи результати дослідження, можна побачити, що найбільш ефективною виявилася структура нейронної мережі на основі декількох лінійних шарів у поєднанні із шаром нормалізації. Вона зарекомендувала себе найкраще у CLIP моделі, що використовує ResNet-50 (максимальне значення косинусної подібності 0.62) та ViT-B/32 (максимальне значення косинусної подібності 0.66) у поєднанні із кодувальником тексту на основі BERT. Експерименти також демонструють, що у випадку, якщо для кодування зображення CLIP використовується ResNet-50, то оптимальним вибором для навчання моделі є NT-Xent, а ViT-B/32 – функція втрат на основі перехресної ентропії. Такі відмінності зумовлені структурними особливостями моделей та різними вихідними розмірами векторів зображень.

Дослідження впливу кількості нейромережевих блоків показало, що поступове зменшення довжини векторного представлення тексту та зображення підвищує точність моделі, а надмірна їхня кількість негативно впливає на збіжність результатів незалежно від функції втрат, що використовується під час тренування моделі. Слід також зауважити, що збільшення кількості блоків збільшує час тренування моделі та час отримання результатів. Отже, під час проєктування архітектури, що використовуватиметься для роботи із векторними представленнями зображень та тексту, завжди варто шукати оптимальний баланс між її складністю та здатністю до узагальнення.

Запропоновані методи можуть бути використані як для підвищення точності CLIP моделей, що використовують BERT як кодувальник тексту, так і для уникнення критичних помилок на етапі проєктування структури нейронної мережі. Проте слід зауважити, що описані у роботі структурні зміни шарів трансформації можуть бути не оптимальними, якщо в CLIP використовуються кодувальники, що не застосовувалися у модифікованих мережах, або на етапі навчання були використані інакші параметри навчання, оптимізатори, функції втрат тощо. Дослідження демонструє, що вивчення різноманітних аспектів CLIP моделі є актуальним завданням, оскільки може сприяти підвищенню її точності та адаптації до конкретних завдань.

References

1. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186. Association for Computational Linguistics. http://doi.org/10.18653/v1/N19-1423.
2. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. & others (2019). Language models are unsupervised multitask learners. OpenAI blog, 1, 9.
3. Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., & Langlotz, C. P. (2022). Contrastive learning of medical visual representations from paired images and text. In Machine Learning for Healthcare Conference (pp. 2-25). PMLR.
4. Sariyildiz, M. B., Perez, J., & Larlus, D. (2020). Learning visual representations with caption annotations. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16 (pp. 153-170). Springer International Publishing.
5. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR.
6. Conde, M. V., & Turgutlu, K. (2021). CLIP-Art: Contrastive pre-training for fine-grained art classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 3956-3960). https://doi.org/10.1109/CVPRW53098.2021.00444
7. Charic D., Farinango C. (2022). Exploring Contrastive Learning for Multimodal Detection of Misogynistic Memes, NAACL 2022, Underline Science Inc. https://doi.org/10.48448/03bw-rm43.
8. Zhou, C., Loy, C. C., & Dai, B. (2022). Extract Free Dense Labels From CLIP. In Computer Vision – ECCV 2022: 17th European Conference (pp. 696–712). Springer-Verlag. https://doi.org/10.1007/978-3-031-19815-1_40.
9. Deuser, F., Habel, K., Rösch, P. J., & Oswald, N. (2022). Less Is More: Linear Layers on CLIP Features as Powerful VizWiz Model. https://doi.org/10.48550/arXiv.2206.05281.
10. Huang, Y., Tang, J., Chen, Z., Zhang, R., Zhang, X., Chen, W., Zhao, Z., Lv, T., Hu, Z., & Zhang, W. (2023). Structure-CLIP: Enhance multi-modal language representations with structure knowledge. https://doi.org/10.48550/arXiv.2305.06152.
11. Li, M., Xu, R., Wang, S., Zhou, L., Lin, X., Zhu, C., Zeng, M., Ji, H., & Chang, S. (2022). CLIP-Event: Connecting text and images with event structures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 16399-16408). https://doi.org/10.1109/CVPR52688.2022.01593.
12. Pei, R., Liu, J., Li, W., Shao, B., Xu, S., Dai, P., Lu, J., & Yan, Y. (2023). CLIPPING: Distilling CLIP-based models with a student base for video-language retrieval. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 18983-18992). https://doi.org/10.1109/CVPR52729.2023.01820.
13. Hyung, J., Hwang, S., Kim, D., Lee, H., & Choo, J. (2023). Local 3D editing via 3D distillation of CLIP knowledge. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 12674-12684). https://doi.org/10.1109/CVPR52729.2023.01219
14. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. https://doi.org/10.48550/arXiv.1409.1556.
15. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770-778). Las Vegas, NV, USA. https://doi.org/10.1109/CVPR.2016.90.
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2010.11929.
17. Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. https://doi.org/10.48550/arXiv.1607.06450.
18. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. https://doi.org/10.48550/arXiv.1207.0580.
19. Sohn, K. (2016). Improved deep metric learning with multi-class N-pair loss objective. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16) (pp. 1857-1865).
20. Schroff, F., Kalenichenko, D. and Philbin, J. (2015) Facenet: A Unified Embedding for Face Recognition and Clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, 7-12 June 2015, 815-823. https://doi.org/10.1109/CVPR.2015.7298682.
21. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. International Journal of Computer Vision (IJCV), 115(3), 211-252. https://doi.org/10.1007/s11263-015-0816-y.
22. Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research, 47, 853-899. https://doi.org/10.1613/jair.3994.
23. Young, P., Lai, A., Hodosh, M., & Hockenmaier, J. (2014). From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics, 2, 67-78. https://doi.org/10.1162/tacl_a_00166.
24. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. https://doi.org/10.48550/arXiv.1412.6980.