Title: Applying Vision Transformers and handle imbalanced NIR data

1. Introduction
   1. Data Description
      * Tóm Tắt:

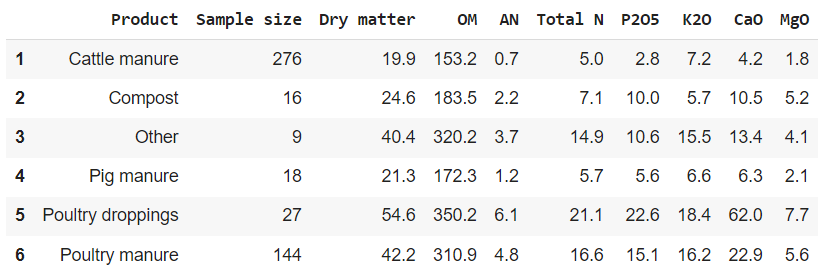
Chất thải hữu cơ động vật rắn (OWPs)[1] từ trang trại chăn nuôi có giá trị phân bón cao (N, P, K), nhưng cũng có thể gây ra các vấn đề về môi trường khi được sử dụng ở lượng quá mức. Do thành phần của chúng biến đổi lớn, việc phát triển các phương pháp nhanh chóng, đáng tin cậy và chi phí thấp để xác định nội dung hóa học của chúng là rất quan trọng. Phổ gần hồng ngoại gần (NIRS) cung cấp khả năng phân tích nhanh chóng các mẫu và không yêu cầu chuẩn bị mẫu nhiều, và các nghiên cứu trước đây đã chứng minh rằng NIRS có thể xác định các tham số cấu thành quan trọng nhất của phân bón động vật rắn. Sự phát triển gần đây của máy quang phổ mini giá thấp đã cho phép thiết bị trải đất có thể trang bị cảm biến để đo thành phần trong thời gian thực, và một số ứng dụng đã được thương mại hóa để phân phối OWPs lỏng. Phân tích in-situ của những sản phẩm vô cùng không đồng nhất này (bề mặt thô, độ ẩm) là một thách thức đối với các ứng dụng như vậy, vì việc thu thập phổ phải thực hiện trên các mẫu thô mà không cần chuẩn bị. Để đánh giá độ chính xác mà NIRS xác định nội dung chất khô, chất hữu cơ, tổng và amoni nitơ, phospho, kali, canxi và magiê, chúng tôi đã tạo ra một cơ sở dữ liệu hiệu chuẩn lớn đại diện cho phân bón động vật rắn thô gặp ở Brittany. Tổng cộng có 490 mẫu phân OWPs động vật rắn từ các trang trại chăn nuôi đã được thu thập vào đầu mùa xuân từ 270 trang trại ở Brittany (Pháp), trong 2 chiến dịch được tiến hành vào năm 2018 và 2019. Mục tiêu của việc thu thập là thu thập mẫu chất thải hữu cơ động vật rắn mà đại diện cho đa dạng về thành phần của các sản phẩm hữu cơ này (chẳng hạn như loài động vật, chế độ chăn nuôi, cách lưu trữ (đống vs. hàng ngang) và tiềm năng phân hủy thành phần hữu cơ.

* + - Mô Tả Dataset:

| Lĩnh vực | Hóa học, Hóa học phân tích: Phổ quang |
| --- | --- |
| Specific subject area | Chemical composition of solid animal manure (cattle, pig and poultry manure; poultry droppings) |
| Loại dữ liệu | Bảng  Hình ảnh |
| Cách dữ liệu được thu thập | Phân tích hóa học được thực hiện cho nội dung chất khô, chất hữu cơ, tổng và amoni nitơ, phospho, kali, canxi và magiê, và tuân theo các phương pháp phân tích được chứng nhận bởi tổ chức tiêu chuẩn Pháp (AFNOR). Các mẫu đã được quét bằng phổ gần hồng ngoại gần (NIRS) để đo thành phần phổ hóa học của chúng. |
| Định dạng dữ liệu | Dữ liệu gốc và dữ liệu đã được phân tích |
| Nguồn dữ liệu | Dữ liệu được thu thập từ các mẫu chất thải hữu cơ động vật rắn lấy từ 270 trang trại ở Brittany, Pháp.  Dữ liệu cũng bao gồm tọa độ GPS của các điểm thu thập mẫu. |
| Khả năng truy cập dữ liệu | Dữ liệu đã được phân tích và sẵn sàng cho việc sử dụng.  Dữ liệu gốc đã được đặt trong kho dữ liệu công cộng có tên "Data INRAE" với số danh hiệu nhận dạng dữ liệu: 10.15454/E1JI8U.  Đường dẫn trực tiếp đến dữ liệu gốc có thể tìm thấy: <https://doi.org/10.15454/E1JI8U> |

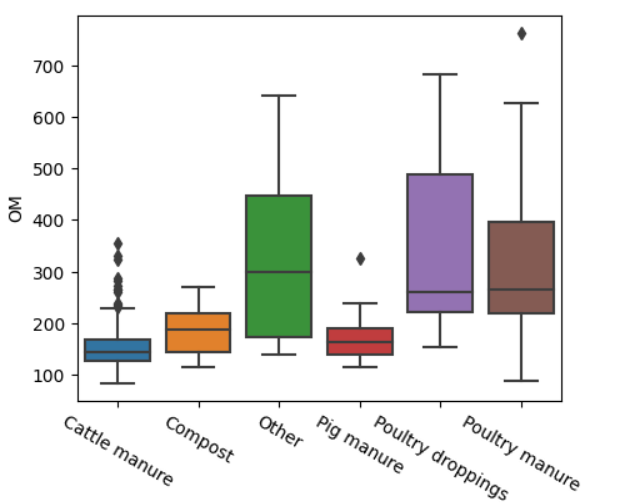
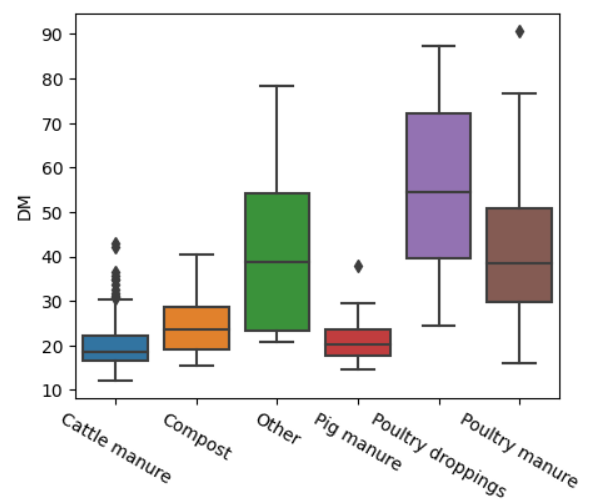
* + - Lý do lựa chọn Dataset:
* Vì nó chứa thông tin về các mẫu phân bón động vật rắn có tính đa dạng và tỷ lệ mẫu bị mất cân bằng lớn. Điều này là quan trọng khi thực hiện các nhiệm vụ liên quan đến việc xử lý dữ liệu không cân bằng và phân tích dữ liệu phức tạp.
* NIRs từ các mẫu thô làm cho nó trở thành một nguồn tài nguyên quý bởi vì nó cho phép việc thu thập dữ liệu từ các mẫu thô mà không cần chuẩn bị mẫu phức tạp.
* Nghiên cứu giá trị phân bón của các sản phẩm hữu cơ và thực hiện các phân tích liên quan đến hóa học, spectroscopy và dữ liệu không cân bằng.
  1. Mô tả chi tiết:’

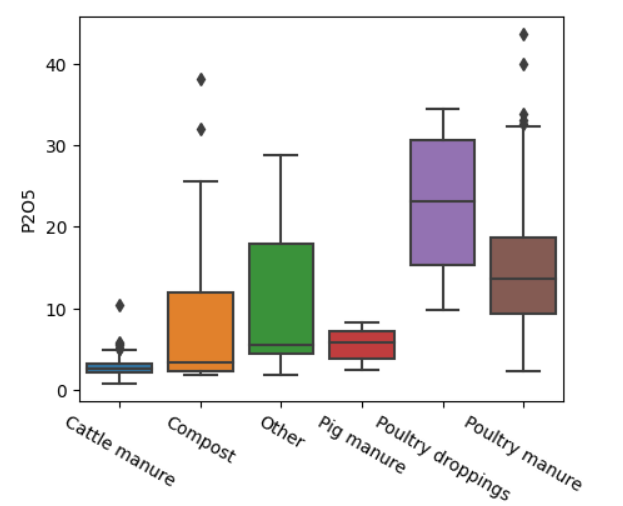
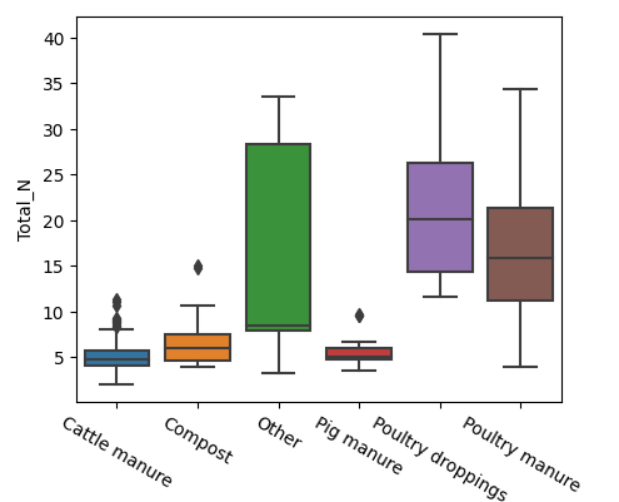
Dữ liệu phổ NIR của 490 chất thải hữu cơ động vật được thu thập, bao gồm giá trị trung bình của các thông số thành phần hóa học theo loại OWP (Bảng 1); Boxplots 6 thông số thành phần hóa học, biểu thị sự thay đổi thành phần theo loại OWP (Hình 1); mối quan hệ giữa chất khô, chất hữu cơ và tổng hàm lượng nitơ (Fig 2); Nội dung của tập dữ liệu (Bảng 2).



Bảng 1: Giá trị trung bình của các thông số thành phần hóa học theo loại chất thải hữu cơ

(DM: dry matter content; OM: organic matter content AN: ammonium nitrogen content; Total N: total nitrogen content P2O5: phosphorus oxide content; K2O: potassium oxide content; CaO: calcium oxide content; MgO: magnesium oxide content).





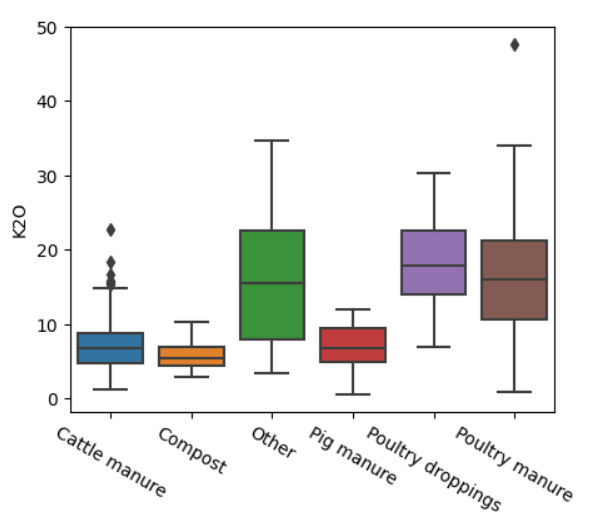
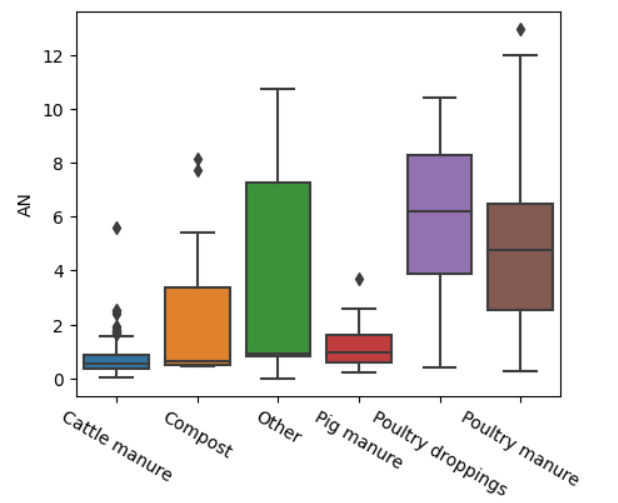


Fig 1: Boxplots (Whiskers represent 1.5 times the interquartile range)

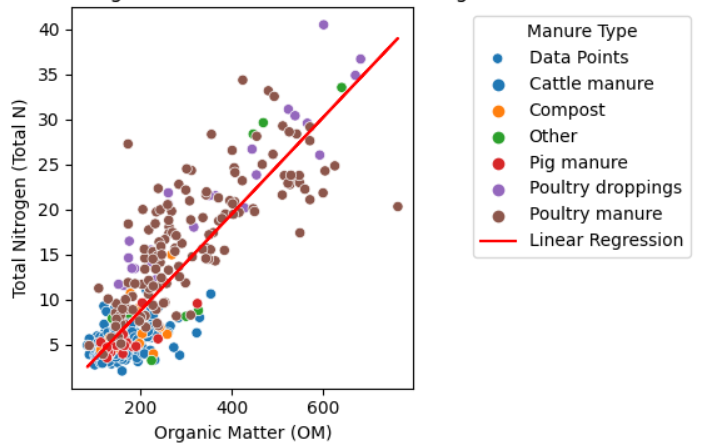
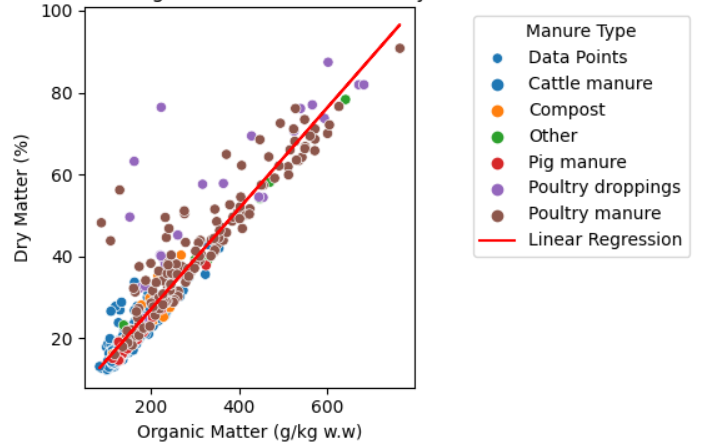


Fig 2: (a) Relation between Organic Matter (OM) and Total Nitrogen (Total N)

(b) Relation between Organic Matter (OM) and Total Nitrogen (Total N)

| File name | Variable name | Content |
| --- | --- | --- |
| chemical\_analysis.xls | Id\_sample  LAT\_WGS84  LONG\_WGS84  Manure\_type  DM  OM  AN  Total\_N  P2O5  K2O  CaO  MgO | Sample identification number  Latitude of the sampling point  Longitude of the sampling point  Manure type  Dry matter content (%)  Organic matter content (g / kg w.w.)  Ammonium nitrogen content (g N / kg w.w.)  Total nitrogen content (g N / kg w.w.)  Total phosphorus oxide content (g P2O5 / kg w.w.)  Total potassium oxide content (g K2O / kg w.w.)  Calcium oxide content (g CaO / kg w.w.)  Magnesium oxide content (g MgO / kg w.w.) |
| spectra.csv |  | Cột đầu tiên chứa số nhận dạng mẫu  Các cột còn lại chứa dữ liệu độ hấp thụ được ghi từ 802–2502 nm |

Bảng 2: Nội dung của tập dữ liệu (w.w. : wet weight)

1. Related Works
   1. Applying ML for NIR data approach
   2. Applying DL for NIR data approach

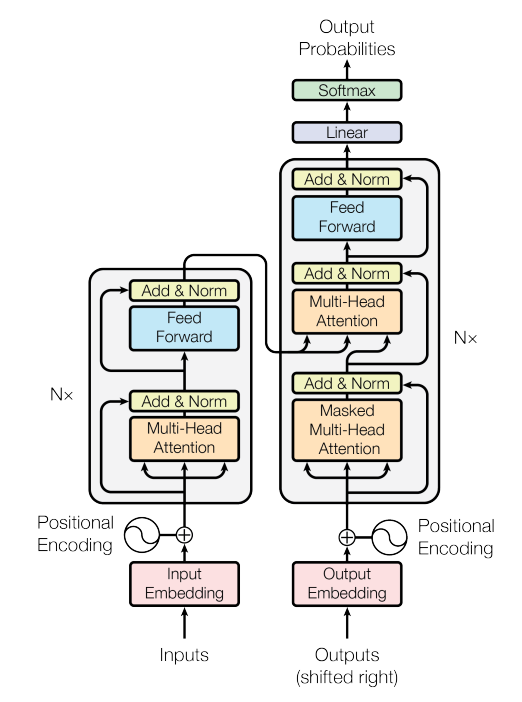
Một số nghiên cứu công bố gần đây [ ] [ ] [ ] [ ] cũng cho thấy rằng các phương pháp deep learning cho ra những kết quả vượt trội [ ].

Jingru Yang et al.

* 1. Imbalanced Data in general computer vision problem
     + Deep learning models are sensitive with imbalanced data -> lower performance
     + Accuracy metric is not fair when evaluate model with imbalanced data
     + …

1. Methodology
2. Vision Transformers
   * 1. Revision of Vanila Transformers

Vanilla Transformer, hay còn gọi là kiến trúc Transformer gốc, là một mô hình mạng neural được giới thiệu trong bài báo năm 2017 có tựa đề "Attention Is All You Need" do Vaswani et al viết. Đây là một trong những bước đột phá quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và đã tạo ra sự cách mạng trong cách xử lý và biểu diễn dữ liệu chuỗi. Nó đã đạt được những kết quả tiên tiến về nhiều nhiệm vụ NLP, bao gồm dịch máy, tóm tắt văn bản và trả lời câu hỏi,... Vanilla Transformer sử dụng kiến trúc dựa trên attention để hiểu và biểu diễn các mối quan hệ giữa các phần tử trong chuỗi mà không cần phụ thuộc vào kiến trúc recurrent neural networks (RNN) hoặc convolutional neural networks (CNN). Dưới đây là kiến trúc mô hình Transformer:



Kiến trúc mô hình Transformers

3.1.1.1 Đầu vào và Nhúng (Embedding):

Mô hình bắt đầu bằng việc biểu diễn các từ hoặc ký hiệu đầu vào thành các vector. Các từ hoặc ký hiệu này được mã hóa bằng các vector nhúng, thường được gọi là embeddings.

Nhúng Đầu Vào(Input Embedding)

Đầu vào: Một token đầu vào được biểu diễn bằng một vector nhúng có kích thước .

Đầu ra: Vector nhúng của token đầu vào là

Công thức:

Trong đó:

là ma trận trọng số cho lớp nhúng đầu ra.

là token đầu ra

3.1.1.2 Mã Hóa (Encoder):

Mô hình sử dụng một ngăn xếp lớp mã hóa, thường là 6 lớp mã hóa trong bản gốc của Transformer. Mỗi lớp mã hóa bao gồm hai phần:

Multi-Head Self-Attention: Cơ chế attention cho phép mô hình tự đánh giá mức độ quan trọng của các từ trong chuỗi đầu vào đối với từng từ. Mỗi lớp mã hóa sử dụng nhiều đầu attention để biểu diễn các mối quan hệ khác nhau. Tuyến tính hóa queries, keys và values lần lượt với các phép chiếu tuyến tính khác nhau, để có keys và kích thước và tương ứng. Sau đó thực hiện hàm tập trung trên các phiên bản đã được chiếu của Queries, keys và values cho kết quả đầu ra có kích thước . Những giá trị này được nối lại và sau đó được chiếu một lần nữa tạo ra các giá trị cuối cùng. Công thức tính như sau:

Ảnh có chứa ảnh chụp màn hình, văn bản, Phông chữ

Mô tả được tạo tự động

Đầu vào: queries(), keys() và values().

Đầu ra: Đầu ra sau Multi-Head Self-Attention ()

Công thức tính Attention Scores:

Trong đó:

là ma trận đầu vào của queries, keys và values.

là kích thước chiều của keys

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Mô tả được tạo tự động

Position-wise Feed-Forward Network: Sau khi thực hiện tập trung tự nhiên, một mạng truyền đổi theo vị trí được áp dụng cho mỗi từ riêng lẻ. Mạng này bao gồm hai biến đổi tuyến tính với một hoạt động ReLU ở giữa.

Đầu vào: Ma trận đầu ra sau tập trung (Attention)

Đầu ra: Đầu ra sau mạng truyền đổi (FFN(x))

Công thức tính đầu ra FFN(x):

Trong đó:

là ma trận đầu vào sau attention

là các ma trận trọng số và vector bias cho các biến đổi tuyến tính

Hai tích chập với kích thước kernel là 1. Kích thước của đầu vào và đầu ra là = 512, và lớp bên trong có kích thước = 2048.

3.1.1.3 Tích Hợp Lớp Mã Hóa và Lớp Giải Mã (Decoder):

Tương tự như lớp mã hóa, mô hình sử dụng một ngăn xếp lớp giải mã, thường là 6 lớp giải mã.

Mỗi lớp giải mã cũng bao gồm hai phần:

Multi-Head Self-Attention: Tương tự như trong lớp mã hóa, nhưng lần này mô hình cần tập trung vào thông tin từ phía trước trong chuỗi đầu ra để duy trì tính tự hồi quy.

Encoder-Decoder Attention: Lớp này cho phép lớp giải mã tập trung vào thông tin ở lớp mã hóa để học cách dịch ngược lại từ ngôn ngữ đích.

Đầu vào: queries() từ lớp giải mã, keys() và values() từ lớp mã hóa.

Đầu ra: Đầu ra sau tập trung mã hóa-giải mã

Công thức tính Attention Scores:

Trong đó:

là ma trận đầu vào của queries, keys và values.

là kích thước chiều của keys

3.1.1.4 Biểu Diễn Đầu Ra và Hàm Softmax:

Đầu ra của lớp nhúng có kích thước (512) và được tổng hợp với biểu diễn vị trí (positional encoding) để bảo toàn thông tin vị trí trong chuỗi.

Cuối cùng, đầu ra từ lớp giải mã được chuyển thành xác suất cho các từ hoặc ký hiệu đầu ra tiếp theo bằng cách sử dụng một lớp biến đổi tuyến tính và hàm softmax.

Nhúng Đầu Ra(Output Embedding)

Đầu vào: Một token đầu ra được biểu diễn bằng một vector nhúng có kích thước

Đầu ra: Vector nhúng của token đầu ra là

Công thức tính:

Trong đó:

là ma trận trọng số cho lớp nhúng đầu ra.

là token đầu ra

Hàm softmax:

Đầu vào: Vector sau lớp nhúng đầu ra

Đầu ra: Xác suất cho các token đầu ra tiếp theo

Công thức:

Trong đó:

là xác suất cho token đầu ra

là phần tử thứ của vector

3.1.1.5 Biểu Diễn Vị Trí (Positional Encoding):

Đầu vào: Vị trí của các từ trong chuỗi.

Đầu ra: Các vector biểu diễn vị trí được thêm vào embeddings.

Công thức tính biểu diễn vị trí:

)

)

Trong đó:

là vị trí của từng chuổi

là kích thước chiều của embeddings

Transformer là một kiến trúc học máy mạnh mẽ đã đạt được kết quả vượt trội trên nhiều nhiệm vụ xử lý ngôn ngữ tự nhiên (NLP). Tuy nhiên, Transformer vẫn còn hạn chế trong việc xử lý các loại dữ liệu khác. Vì vậy, Mô hình Vision Transformer (ViT) đã được giới thiệu vào năm 2021 trong một bài báo nghiên cứu được xuất bản dưới dạng báo cáo hội nghị tại ICLR 2021, có tiêu đề “An Image is Worth 16\*16 Words: Transformers for Image Recognition in Scale”. Nó được phát triển và xuất bản bởi Neil Houlsby, Alexey Dosovitskiy, và 10 tác giả khác của Google Research Brain Team. Biến thể ViT đã nhanh chóng trở nên phổ biến, đạt được kết quả vượt trội so với các kiến trúc xử lý hình ảnh truyền thống như convolutional neural network (CNN).

* + 1. Vision Transformers Architecture

Vision Transformers (ViT) là một kiến trúc mạng nơ-ron sử dụng trong lĩnh vực thị giác máy tính, và nó đã được giới thiệu vào năm 2021 bởi các nhà nghiên cứu của Google Research. Vision Transformers khác biệt với các kiến trúc thị giác trước đây như Convolutional Neural Networks (CNNs) bằng cách sử dụng transformer architecture, kiến trúc được sử dụng ban đầu cho xử lý ngôn ngữ tự nhiên. Kiến trúc của mô hình gồm 3 thành phần chính: Linear Projection of Flattened Patches, Transformer encoder và Classification head.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ

Mô tả được tạo tự động

Mô tả cơ bản về kiến trúc Vision Transformer

* + - 1. Linear Projection of Flattened Patches

Patch Embedding:

Ảnh đầu vào có kích thước () trong đó và là chiều cao và chiều rộng, là số kênh màu. Ảnh đầu vào sẽ được chia nhỏ các patch với kích thước () sau đó đưa các patches này về dạng vector bằng cách flattend các patches này ra.

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, biểu đồ

Mô tả được tạo tự động

Hình này mô tả phần Linear Projection. Nó là một lớp Dense với đầu vào flattend vector của các patches, đầu ra là embedding vector tương ứng của từng patch.

Công thức tính:

Trong đó:

là flattend vector của patch thứ .

là output tương ứng của khi qua Linear Projection.

được gọi là ma trận embeeding.

Positional Embeeding:

Tương tự với mô hình Transformer gốc. Positional embeeding trong mô hình ViT sẽ chứa thông tin về vị trí của patch trong ảnh (spatial information). Sau khi có vector positional embeeding cho mỗi patch chúng ta sẽ cộng các vector này tương ứng với embeeding vector của từng patch đã tính ở trên và thu được các vector embeeding vừa chứa thông tin của vùng ảnh vừa chứa thông tin về vị trí của nó trong ảnh.

Bước 1: Tính Positional Embeeding:

Positional Embedding cho patch thứ là một vector gồm 2 thành phần là và

Công thức:

)

)

Trong đó:

là vị trí của từng chuổi

là kích thước chiều của embeddings

Bước 2: Cộng Positional Embedding vào vector embedding của từng patch thứ

Trong đó,

là vector embedding của patch thứ (đã tính bằng công thức )

là vector embedding cuối cùng cho patch thứ , chứa thông tin về cả vùng ảnh và vị trí của patch trong ảnh.

Class Embedding: là một vector chứa các thông tin về toàn bộ hình ảnh và thường được ký hiệu là

Công thức tính class embedding cho hình ảnh đầu vào là:

Trong công thức trên:

là vector embedding cuối cùng cho toàn bộ hình ảnh (bao gồm thông tin từ Class Embedding và thông tin từ tất cả các patch).

C là vector embedding đại diện cho lớp (class) của hình ảnh.

là vector embedding cuối cùng cho patch thứ , đã chứa thông tin về vùng ảnh và vị trí cụ thể của patch đó trong ảnh sau khi đã cộng Positional Embedding.

là vector Positional Embedding cho patch thứ .

là tổng của vector embedding của tất cả các patch sau khi đã cộng Positional Embedding.

* + - 1. Transformer Encoder

Một lớp Transformer Encoder trong ViT bao gồm nhiều khối giống nhau, thường được gọi là "blocks" hoặc "layers." Mỗi khối này bao gồm hai phần chính: Multi-Head Self-Attention Layer và Feedforward Neural Network Layer

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Kiến trúc của Transformer Encoder

Self Attention Layer: là thành phần chính để tạo nên một block trong Transformer EncoderẢnh có chứa văn bản, hàng, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Đầu vào của Self attention layer là một chuỗi

Đầu ra của Self Attention layer là một context vector chứa những thông tin quan trọng nhất của chuỗi đầu vào

Các parameters của layer này bao gồm:

Multi-Head Self-Attention Layer: dùng để học các mối quan hệ không gian giữa các phần của hình ảnh. Đầu vào cho lớp này là vector embedding của toàn bộ hình ảnh, gọi là , cùng với vector embedding của các patch, (bao gồm thông tin từ Positional Embedding và Class Embedding).

Mỗi khối thường có nhiều đầu tự chú ý (multi-head self-attention heads) để học các mối quan hệ không gian khác nhau.

Ảnh có chứa văn bản, hàng, biểu đồ, ảnh chụp màn hình

Mô tả được tạo tự động

Công thức tổng quát cho Self-Attention cho một head:

Trong đó:

là ma trận đầu vào của queries, keys và values.

là kích thước chiều của keys

Feedforward Neural Network Layer: Sau khi đã tính toán tổng hợp thông tin thông qua Self-Attention Layer, ta cần biến đổi lại thông tin đó thông qua một lớp Feedforward Neural Network.

Công thức tổng quát cho lớp này:

Trong đó:

là đầu vào (đã qua Self-Attention)

là ma trận trọng số và bias của tầng đầu tiên,

là ma trận trọng số và bias của tầng thứ hai.

Mỗi lớp Transformer Encoder trong mô hình ViT bao gồm một lớp Multi-Head Self-Attention để học các mối quan hệ không gian và một lớp Feedforward Neural Network để biến đổi thông tin. Các lớp này hoạt động cùng nhau để học biểu diễn của hình ảnh và vị trí của các patch trong ảnh. Mô hình ViT thường có nhiều lớp Transformer Encoder, và các lớp này được xếp chồng lên nhau để tạo ra biểu diễn đầy đủ cho hình ảnh.

* + - 1. Classification Head

Sau khi thông tin đã được xử lý qua một số lớp Transformer, một lớp phân loại (classification head) được thêm vào để dự đoán nhãn của ảnh đầu vào. Thường thì là một lớp Fully Connected với hàm softmax ở đầu để tính xác suất cho các lớp phân loại

Giả sử có một tập hợp các lớp phân loại, được đánh số từ 1 đến , với là số lượng lớp. Khi thông tin đã được xử lý qua các lớp Transformer và đã có một biểu diễn cuối cùng của ảnh (thường là một vectơ), sử dụng lớp phân loại để dự đoán xác suất thuộc về từng lớp. Dưới đây là công thức:

Đầu vào của lớp phân loại (còn gọi là ) được tính bằng cách nhân biểu diễn cuối cùng của ảnh với ma trận trọng số và cộng thêm vectơ bias :

Trong đó:

là vectơ đầu ra của lớp phân loại.

là biểu diễn cuối cùng của ảnh, thường là một vectơ sau khi thông qua lớp Transformer Encoder (bao gồm thông tin từ Class Embedding và Positional Embedding).

là ma trận trọng số của lớp phân loại.

là vectơ bias của lớp phân loại.

Sau khi tính được , chúng ta sử dụng hàm softmax để chuyển chúng thành các xác suất thuộc về từng lớp:

Trong đó:

) là xác suất ảnh thuộc về lớp .

là hàm softmax, chuyển đổi thành các xác suất.

là thành phần thứ trong vectơ .

1. Imbalanced Data
   * 1. Focal Loss

Focal Loss (Hàm Mất Mát Focal) là một hàm mất mát được thiết kế để xử lý vấn đề mất cân bằng lớp (class imbalance) trong các nhiệm vụ phân loại trong học máy và học sâu.

Sử dụng Focal Loss giúp giảm thiểu tác động của mất cân bằng lớp và tập trung vào việc cải thiện khả năng phân loại cho các dữ liệu mất cân bằng. Điều này có thể dẫn đến việc huấn luyện mô hình có độ hiệu suất tốt hơn trên các lớp thiểu số và tỷ lệ false positive (dự đoán sai) thấp hơn.

Hàm mất mát Focal Loss được tính bằng công thức sau đây:

Trong đó:

là xác suất dự đoán của mô hình cho lớp đúng.

​ là một hệ số cân bằng để kiểm soát đóng góp của từng lớp.

là một tham số để điều chỉnh độ tập trung vào các dữ liệu phức tạp hơn.

* + 1. Generative Method
    2. …

1. Experimental Result
   1. Dataset (description)
   2. Data preprocessing (preprocessing pipeline)
   3. Experimental result
      1. Implementation detail
         1. Hardware
         2. Framework & environment
         3. Training strategy
      2. Metrics & Evaluation
         1. Metrics: precision, recall, f1 score, auc roc

Dựa trên các kỹ thuật xử lý dữ liệu và phương pháp mô hình thực nghiệm được đề xuất ở trên.Các loại phân bón đã được phân loại và xác định. Đầu tiên chúng tôi thử nghiệm phân loại phân bón dựa trên các mô hình MLP, TeaNet, TeaResNet kết hợp với hàm mất mát Focal Loss với các phương pháp đánh giá Precision, Recall, F1-Score, AUC ROC. Dưới đây là một số hình ảnh kết quả đánh của các mô hình.

| Method | Training Accuracy | Testing Accuracy | Precision | Recall | F1-Score |
| --- | --- | --- | --- | --- | --- |
| MLP + Focal Loss | 97.67% | 95.92% | 0.969984 | 0.818346 | 0.857465 |
| TeaNet + Focal Loss | 92.71% | 94.56% | 0.981188 | 0.718346 | 0.775476 |
| TeaResNet + Focal Loss | 97.08% | 97.28% | 0.988671 | 0.822222 | 0.873858 |
| ViT | 99.71% | 96.60% | 0.9697 | 0.9660 | 0.9670 |
| ViT + Focal Loss | 99.42% | 95.24% | 0.9607 | 0.9524 | 0.9557 |

* + - 1. Evaluation
  1. Discussion

1. Demonstration (source code)
2. Conclusion
3. References:

[1]<https://entrepot.recherche.data.gouv.fr/dataset.xhtml?persistentId=doi:10.15454/E1JI8U&version=1.0>