

NGÀNH CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN I

TENSORGRAM

PHẦN MỀM TẠO DIAGRAM

CHO MÔ HÌNH TENSORFLOW

|  |  |
| --- | --- |
| Nhóm sinh viên thực hiện: | |
| Huỳnh Quốc Hoàng Vương | 17110256 |
| Nguyễn Minh Trung  Việt | 17110xxx  17110xxx |

GVHD: TS. Huỳnh Xuân Phụng

Tp. Hồ Chí Minh, tháng 11 – 2019

ĐIỂM SỐ

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Giáo viên hướng dẫn

(*ký và ghi họ tên*)

Huỳnh Xuân Phụng

ĐIỂM SỐ

|  |  |  |  |
| --- | --- | --- | --- |
| TIÊU CHÍ | NỘI DUNG | TRÌNH BÀY | TỔNG |
| ĐIỂM |  |  |  |

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Giáo viên phản biện

(*ký và ghi họ tên*)

LỜI CẢM ƠN

Để hoàn thành tốt đề tài và bài báo cáo này, chúng em xin gửi lời cảm ơn chân thành đến giảng viên, tiến sĩ Huỳnh Xuân Phụng, người đã trực tiếp hỗ trợ chúng em trong suốt quá trình làm đề tài. Chúng em cảm thầy đã đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình để định hướng cho chúng em đi đúng với yêu cầu của đề tài đã chọn, luôn giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp chúng em khắc phục nhược điểm và hoàn thành tốt cũng như đúng thời hạn Khoa đã đề ra.

Chúng em cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Đào tạo Chất Lượng Cao nói chung và nghành Công Nghệ Thông Tin nói riêng đã tận tình truyền đạt những kiến thức cần thiết giúp chúng em có nền tảng để làm nên đề tài này, đã tạo điều kiện để chúng em có thể tìm hiểu và thực hiện tốt đề tài. Cùng với đó, chúng em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp chúng em có thể hoàn thiện hơn đề tài của mình.

Đề tài và bài báo cáo được chúng em thực hiện trong khoảng thời gian ngắn, với những kiến thức còn hạn chế cùng nhiều hạn chế khác về mặt kĩ thuật và kinh nghiệm trong việc thực hiện một dự án phần mềm. Do đó, trong quá trình làm nên đề tài có những thiếu sót là điều không thể tránh khỏi nên chúng em rất mong nhận được những ý kiến đóng góp quý báo của các quý thầy cô để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Chúng em xin chân thành cảm ơn.

Cuối lời, chúng em kính chúc quý thầy, quý cô luôn dồi dào sức khỏe và thành công hơn nữa trong sự nghiệp trồng người. Một lần nữa chúng em xin chân thành cảm ơn.

TP.HCM, ngày 10 tháng 12 năm 2018

Nhóm sinh viên thực hiện

MỤC LỤC

[DANH MỤC CÁC HÌNH 1](#_Toc22929485)

[DANH MỤC CÁC BẢNG 2](#_Toc22929486)

[CHƯƠNG 1: TỔNG QUAN CHƯƠNG TRÌNH 3](#_Toc22929487)

[1. Giới thiệu chung 3](#_Toc22929488)

[1.1. Machine Learning, Tensorflow và Layers API 3](#_Toc22929489)

[1.1.1. Lí thuyết Machine Learning cơ bản 3](#_Toc22929490)

[1.1.2. Thư viện ML Tensorflow 3](#_Toc22929491)

[1.1.3. Layers API của Tensorflow 4](#_Toc22929492)

[*1.1.3.1.* *Artificial Neural network (ANN)* 4](#_Toc22929493)

[1.2. Về đồ án 5](#_Toc22929494)

[2. Đặc tả phần mềm TensorGram 5](#_Toc22929495)

[2.1. Phần mềm TensorGram 5](#_Toc22929496)

[2.1.1. Giới thiệu về phần mềm TensorGram 5](#_Toc22929497)

[2.1.2. Use Case Diagram 5](#_Toc22929498)

[2.1.3. Tính năng chính 5](#_Toc22929499)

[2.1.4. Ứng dụng 5](#_Toc22929500)

[2.2. Giao diện chương trình 6](#_Toc22929501)

[2.3. Yêu cầu kĩ thuật 6](#_Toc22929502)

[2.4. Mục tiêu thực hiện 6](#_Toc22929503)

[CHƯƠNG 2: KẾ HOẠCH THỰC HIỆN 7](#_Toc22929504)

[1. Kế hoạch 7](#_Toc22929505)

[2. Phân công công việc 7](#_Toc22929506)

[CHƯƠNG 3: THIẾT KẾ PHẦN MỀM 8](#_Toc22929507)

[1. Thiết kế lớp 8](#_Toc22929508)

[1.1. Thiết kế lớp cho các Layer của TensorFlow Layers API 8](#_Toc22929509)

[1.1.1. Tổng quan 8](#_Toc22929510)

[1.1.2. Thiết kế 9](#_Toc22929511)

[1.1.3. Đặc tả lớp 10](#_Toc22929512)

[1.1.4. Đặc tả các phương thức trong lớp 11](#_Toc22929513)

[1.1.4.1. Lớp Layer 11](#_Toc22929514)

[1.2. Thiết kế lớp chức năng 12](#_Toc22929515)

[*2.* *Thuật toán* 12](#_Toc22929516)

[*3.* *Thiết kế giao diện* 12](#_Toc22929517)

[CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ 13](#_Toc22929518)

[CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 14](#_Toc22929519)

[1. Kết luận 14](#_Toc22929520)

[2. Hướng phát triển 14](#_Toc22929521)

[TÀI LIỆU THAM KHẢO 15](#_Toc22929522)

# DANH MỤC CÁC HÌNH

# DANH MỤC CÁC BẢNG

CHƯƠNG 1: TỔNG QUAN CHƯƠNG TRÌNH

1. *Giới thiệu chung*
   1. Machine Learning, Tensorflow và Layers API
      1. Lí thuyết Machine Learning cơ bản

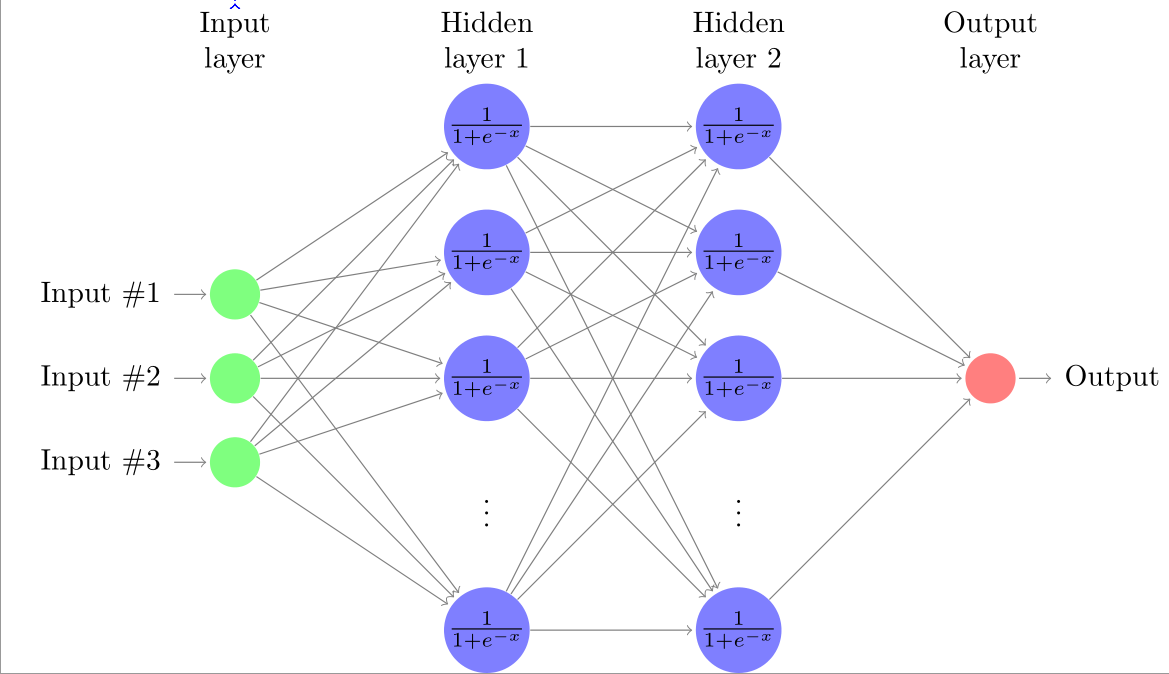
Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư. Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, trợ lí ảo Alexa của Amazon, Cortana của Microsoft, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, …, chỉ là một vài trong vô vàn những ứng dụng của AI.

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia, Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

* + 1. Thư viện ML Tensorflow

TensorFlow là một thư viện Machine Learining được Google phát triển và phát hành vào tháng 10 năm 2015. Thư viện này hỗ trợ xây dựng các mô hình Machine Learning rất phức tạp qua những API cực kì ngắn gọn. Các mô hình Machine Learning phát triển trên TensorFlow có thể được sử dụng trên nhiều nền tảng khác nhau (từ Smartphone tới Distributed Servers) và trên cả CPUs lẫn GPUs.

* + 1. Layers API của Tensorflow
       1. *Artificial Neural network (ANN)*



Mạng Neural nhân tạo là sự kết hợp của của các tầng perceptron hay còn được gọi là perceptron đa tầng.

Một mạng ANN sẽ có 3 kiểu tầng:

* Tầng vào (Input layer): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
* Tầng ra (Output layer): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
* Tầng ẩn (Hidden layer): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng).

Một ANN chỉ có 1 tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn. Trong mạng NN, mỗi nút mạng là một node đơn lẻ nhưng chức năng của chúng có thể khác nhau. Tuy nhiên trong thực tế người ta thường để chúng cùng dạng với nhau để tính toán cho thuận lợi. Ở mỗi tầng, số lượng các nút mạng có thể khác nhau tuỳ thuộc vào bài toán và cách giải quyết.

* + - 1. *Layers API*

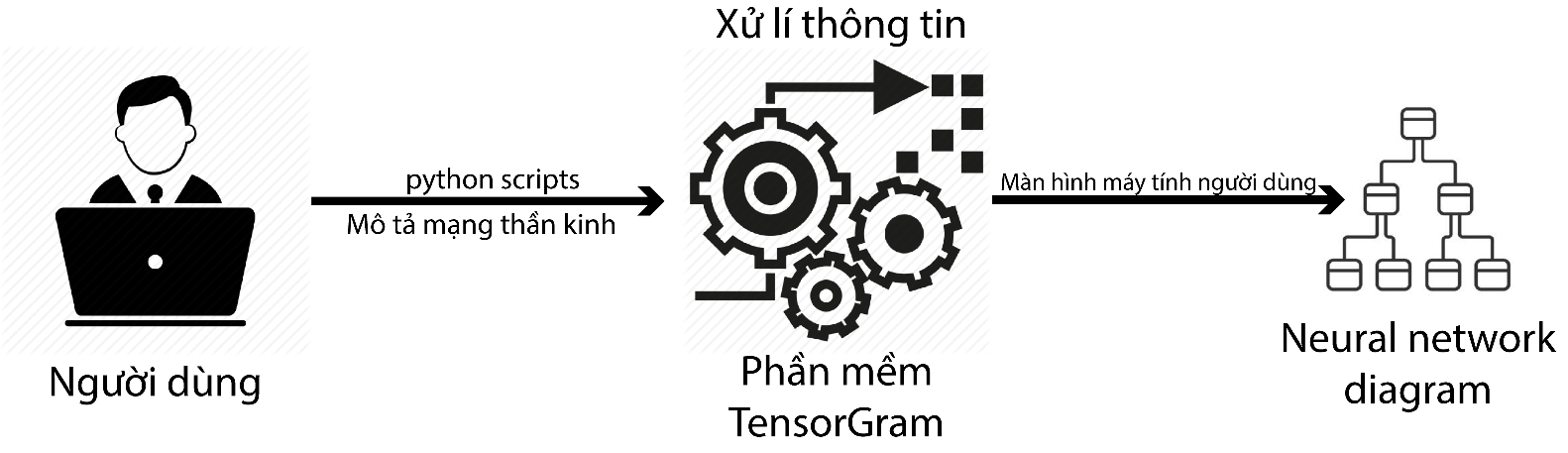
Layers API là một module của TensorFlow, được tạo ra bởi François Chollet, tác giả của bộ thư viện Keras với chức năng tương tự. Layers API được dùng để tạo ra một ANN

* 1. Về đồ án

1. *Đặc tả phần mềm TensorGram*
   1. Phần mềm TensorGram
      1. Giới thiệu về phần mềm TensorGram

TensorGram là một phần mềm nhỏ gọn (portable) chạy mà không cần cài đặt dùng để đồ thị hoá một *ANN* được định nghĩa bởi người dùng thông qua đoạn Python Scripts nhập trực tiếp vào khung soạn thảo trong chương trìnhthành dạng TensorFlow Layers và hiển thị lên cho người dùng.

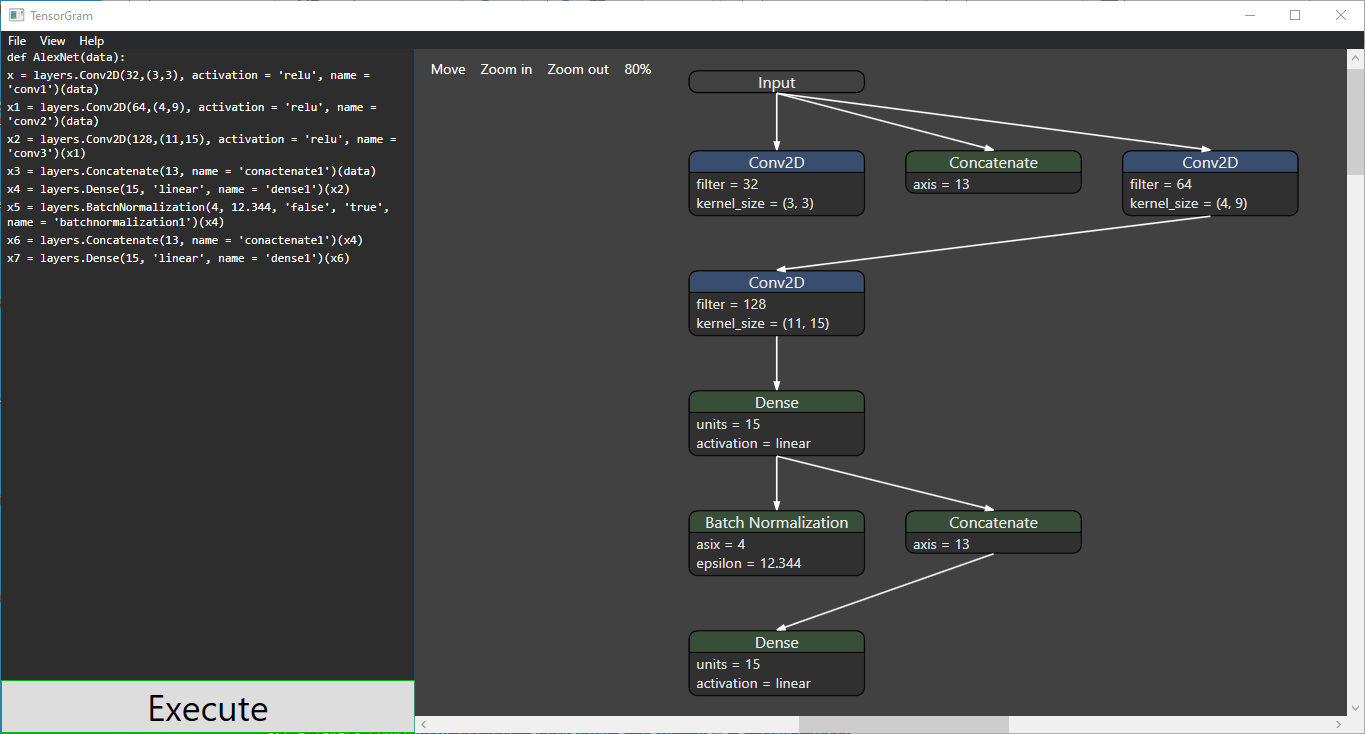
* + 1. Use Case Diagram



* Input:
* Output:
  + 1. Tính năng chính
* Tạo Diragram về ANN dưới dạng TensorFlow Layer và hiển thị trên người dùng.
* Thể hiện thông tin chi tiết về từng Layer trong mạng khi người dùng click chuột vào Layer đó.
* Tìm kiếm Layer theo tên.
  + 1. Ứng dụng

Giúp người dùng có cái nhìn trực quan về ANN của mình dưới dạng TensorFlow Layer Diagram mà không cần phải cài đặt và sử dụng TensorBoard cồng kềnh và phức tạp, cũng như thay vì phải xây dựng lại đầy đủ một ANN thì giờ đây người dùng chỉ cần vài câu python scripts là đủ cho mình một ANN on-the-go.

* 1. Giao diện chương trình



Thanh công cụ

Nút thực thi python scripts 🡪 Diagram

Nơi nhập python scripts chứa mô tả về ANN

Nơi hiển thị TensorFlow Layer Diagram

* 1. Yêu cầu kĩ thuật
  2. Mục tiêu thực hiện

CHƯƠNG 2: KẾ HOẠCH THỰC HIỆN

1. *Kế hoạch*

|  |  |
| --- | --- |
| Tuần | Công việc |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |

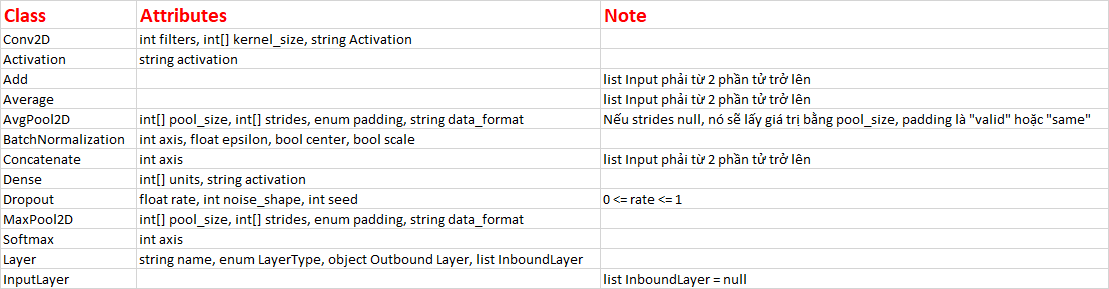
1. *Phân công công việc*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên sinh viên | Miêu tả công việc | Đóng góp |
| 1 | Huỳnh Q.H. Vương |  |  |
| 2 | Minh Trung |  |  |
| 3 | Việt |  |  |

CHƯƠNG 3: THIẾT KẾ PHẦN MỀM

1. *Thiết kế lớp*
   1. Thiết kế lớp cho các Layer của TensorFlow Layers API
      1. Tổng quan

Xem xét trong phạm vi phần mềm sẽ xây dựng, ngoài các class đặc thù phục vụ cho các chức năng và sự vận hành của chương trình, do sự hạng chế về thời gian và kiến thức của chúng em, phần mềm này chỉ cài đặt và hiện thực hoá 13 trong tổng số 203 Layer trong Layers API, cũng như chúng em sẽ chỉ cài đặt cho các layer này các tham số quan trọng chứ không cài đặt toàn bộ tất cả tham số cho chúng.

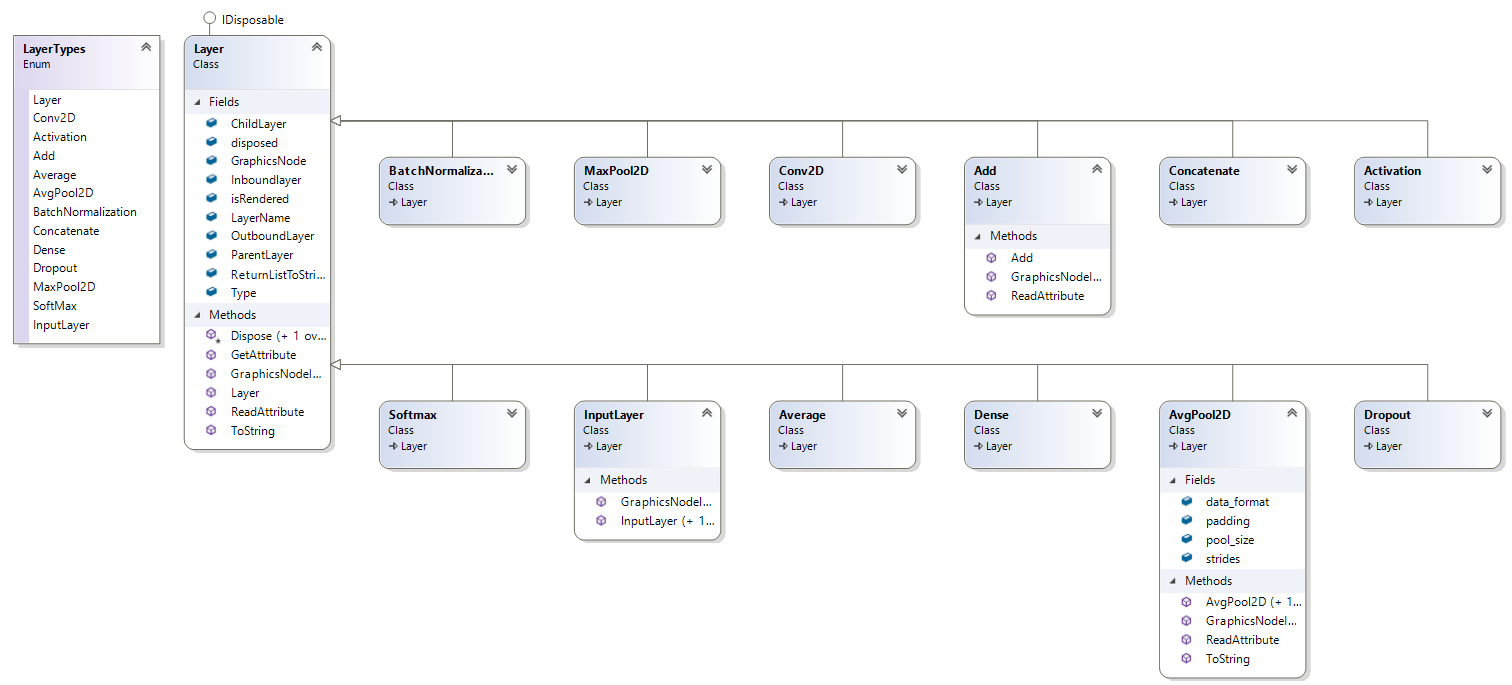


*Hình 3: Các Class và Attributes ứng với các layer.*

|  |  |  |
| --- | --- | --- |
| TT | Tên lớp | Chức năng |
| 1 | Layer | Base layer, là dạng chung của tất cả các class của Keras Layers trong mô hình TensorFlow. |
| 2 | InputLayer | Dùng làm điểm vào cho mạng (Biểu đồ các Layer). |
| 3 | Conv2D | Tạo ra một *convolution kernel,* kết hợp với Input của Layer, tạo ra một Tensor đầu ra. |
| 4 | Activation | Áp dụng hàm activation vào một Output. |
| 5 | Add | Thực hiện phép cộng tất cả các phần tử đồng dạng trong Input, cho ra Output duy nhất 1 phần tử(Đồng dạng với Input) |
| 6 | Average | Thực hiện phép trung bình tất cả các phần tử đồng dạng trong Input, cho ra Output duy nhất 1 phần tử(Đồng dạng với Input) |
| 7 | AvgPool2D | Thực hiện phép trung bình gộp lên dữ liệu Spatial |
| 8 | BatchNormalization |  |
| 9 | Concatenate |  |
| 10 | Dense |  |
| 11 | Propout |  |
| 12 | MaxPool2D |  |
| 13 | Softmax |  |

*Bảng 2: Chi tiết chức năng các layer trong TensorFlow Model*

* + 1. Thiết kế



*Hình 4 : UML Diagram biểu diễn các lớp sẽ xây dựng trong phần mềm.*

Xét thấy giữa các layer có các tính chất chung, bao gồm:

* + Tên layer.
  + Loại layer.
  + Input layer.
  + Output layer.

Đó là chưa kể giữa chúng còn có các methods và chứa các đối tượng giống nhau (ví dụ như method đọc input, đối tượng đồ hoạ, overdrive method .ToString(), …). Vì thế, để tận dụng sức mạnh của lập trình hướng đối tượng, chúng em quyết định thiết kế class Layer, class này là base cho các class khác, chứa tất cả những phương thức, thuộc tính chung của các layer. Đồng thời, các class của các layer khác kế thừa nó, bổ sung các Method, đối tượng của riêng nó để đáp ứng chức năng, nhiệm vụ của layer mà class đó hiện thực.

* + 1. Đặc tả lớp

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Mục đích | SV phụ trách |
| 1 | ILayer | Chứa các hàm dựng cần thiết cho class Layer | Hoàng Vương |
| 2 | Layer  Kế thừa từ: ILayer | Base class, đối tượng hoá Layer *Layer* trong TensorFlow Model. Chứa tất cả những phương thức, thuộc tính chung của các Layer, cho các class khác sử dụng. | Hoàng Vương |
| 3 | InputLayer  Kế thừa từ: Layer | Đối tượng hoá Layer *InputLayer* trong TensorFlow Model. | Hoàng Vương |
| 4 | Conv2D  Kế thừa từ: Layer | Đối tượng hoá Layer *Conv2D* trong TensorFlow Model. | Hoàng Vương |
| 5 | Activation  Kế thừa từ: Layer | Đối tượng hoá Layer *Activation* trong TensorFlow Model. |  |
| 6 | Add  Kế thừa từ: Layer | Đối tượng hoá Layer *Add* trong TensorFlow Model. | Hoàng Vương |
| 7 | Average  Kế thừa từ: Layer | Đối tượng hoá Layer *Average* trong TensorFlow Model. |  |
| 8 | AvgPool2D  Kế thừa từ: Layer | Đối tượng hoá Layer *AveragePooling2D* trong TensorFlow Model. | Hoàng Vương |
| 9 | BatchNormalization  Kế thừa từ: Layer | Đối tượng hoá Layer *BatchNormalization* trong TensorFlow Model. |  |
| 10 | Concatenate  Kế thừa từ: Layer | Đối tượng hoá Layer *Concatenate* trong TensorFlow Model. |  |
| 11 | Dense  Kế thừa từ: Layer | Đối tượng hoá Layer *Dense* trong TensorFlow Model. |  |
| 12 | Propout  Kế thừa từ: Layer | Đối tượng hoá Core *Layer* Propout trong TensorFlow Model. |  |
| 13 | MaxPool2D  Kế thừa từ: Layer | Đối tượng hoá Layer *MaxPool2D* trong TensorFlow Model. |  |
| 14 | Softmax  Kế thừa từ: Layer | Đối tượng hoá Layer *Softmax* trong TensorFlow Model. |  |

*Bảng 3: Danh mục các lớp*

* + 1. Đặc tả các phương thức trong lớp
       1. *Lớp Layer*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file, stt dòng khai báo |
| 1 | virtual ReadAttribute(string \_input)  input: \_input  output: None | Phương thức virtual, dùng để đọc plain text đã qua xử lí từ input của user thành dữ liệu, đưa vào các attributes của layer. | Layers/Topology/Layer.cs (73) |
| 2 | GetAttribute()  Input: None  output: List<string> | Phương thức virtual, dùng để xuất ra tất cả các thông tin về attributes của layer dưới dạng List<string> , mỗi phần tử trong list chưa tên và giá trị của nó. | Layers/Topology/Layer.cs (84) |
| 3 | ToString()  Input: None  output: List<string> | Phương thức virtual, dùng để xuất ra tất cả các thông tin về layer dưới dạng List<string> , mỗi phần tử trong list chưa tên và giá trị của nó. | Layers/Topology/Layer.cs (89) |
| 4 | GraphicsNodeInitialize()  Input: None  output: None | Phương thức virtual, khởi tạo đối tượng đồ hoạ cho layer với tên của đối tượng bằng với tên lớp đọc từ input người dùng | Layers/Topology/Layer.cs (79) |
| 5 | Layer()  Input: None  output: None | Khới tạo một class Layer mới, tất cả các attribute và properties cơ bản được khởi tạo về Null | Layers/Topology/Layer.cs (40) |

* + - 1. *Lớp InputLayer*
      2. *Lớp Conv2D*
      3. *Lớp Activation*
      4. *Lớp Add*
      5. *Lớp Average*
      6. *Lớp AvgPool2D*
      7. *Lớp BatchNormalization*
      8. *Lớp Concatenate*
      9. *Lớp Dense*
      10. *Lớp Propout*
      11. *Lớp MaxPool2D*
      12. *Lớp SoftMax*
  1. Thiết kế lớp chức năng
     1. Tổng quan

Ngoài các lớp chủ đạo của TensorFlow Layers API ra, để chương trình có thể hoạt động được, cần xây dựng các lớp chức năng cho chương trình. Các lớp chức năng này sẽ đảm nhiệm các công việc gồm:

* Đọc, chuẩn hoá và xử lí input từ người dùng.
* Điểu kiển, kiểm soát việc dựng các đối tượng đồ hoạ lên màn hình người dùng.
  + 1. Đặc tả lớp

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Mục đích | SV phụ trách |
| 1 | InputHander | * Đọc đoạn Python Scripts chứa mô tả về ANN bằng Layers API mà người dùng nhập vào theo từng dòng(Mỗi dòng chứa thông tin cho Layer) * Chuẩn hoá, xử lí, tạo object layer ứng với layer được miêu tả, truyền vào thông số chính như loại layer, tên layer, đầu vào, đầu ra, các thuộc tính(dưới dạng plain text thô, chưa qua xử lí) và thêm layer đó vào Model | Hoàng Vương |
| 2 | TensorModel | Tạo ra một model ANN hoàn chỉnh bằng Layers API | Hoàng Vương |
| 3 | Render\_MasterControl | Điều khiển chung toàn bộ các tác vụ liên quan đến đồ hoạ và dựng hình trong chương trình | Hoàng Vương |
| 4 | SlidePanel\_Control | Điều khiển riêng, đặc biệt tới các tác vụ liên quan đến đồ hoạ và dựng hình trong chương trình của đối tượng SlidePanel | Hoàng Vương |
| 5 | ConnectorRender\_Control | Điều khiển riêng, đặc biệt tới các tác vụ liên quan đến đồ hoạ và dựng hình trong chương trình các đối tượng liên kết giữa các layer |  |
| 6 | Arrow | Tạo ra đối tượng đồ hoạ dạng mũi tên | Hoàng Vương |

* + 1. Đặc tả các phương thức trong lớp
       1. *Lớp InputHander*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file, stt dòng khai báo |
| 1 | Render\_MasterControl(Canvas \_maincanvas, TensorModel \_model)  input: \_treelevel  output: None | Phương thức khởi tạo của đối tượng ConnectorRender\_Control | RenderControl/  Render\_MasterControl.cs (30) |
| 2 | void GetParent(ref List<Layer> \_Layers)  Input: \_layer  output: | Phương thức virtual, dùng để xuất ra tất cả các thông tin về attributes của layer dưới dạng List<string> , mỗi phần tử trong list chưa tên và giá trị của nó. | RenderControl/  Render\_MasterControl.cs (50) |
| 3 | void GetChild(ref List<Layer> \_Layers)  Input: \_Layers  output: | Phương thức virtual, dùng để xuất ra tất cả các thông tin về layer dưới dạng List<string> , mỗi phần tử trong list chưa tên và giá trị của nó. | RenderControl/  Render\_MasterControl.cs (38) |
| 4 | void LayerRender(TensorModel \_model)  Input: \_model  output: None | Phương thức virtual, khởi tạo đối tượng đồ hoạ cho layer với tên của đối tượng bằng với tên lớp đọc từ input người dùng | RenderControl/  Render\_MasterControl.cs (63) |
| 5 | protected void SetPosition(int CurrentLevel, ref List<Layer> \_layers, Canvas DisplayZone)  Input: None  output: None | Khới tạo một class Layer mới, tất cả các attribute và properties cơ bản được khởi tạo về Null | RenderControl/  Render\_MasterControl.cs (109) |

* + - 1. *Lớp TensorModel*
      2. *Lớp Render\_MasterControl*
      3. *Lớp SlidePanel\_Control*
      4. *Lớp ConnectorRender\_Control*
      5. *Lớp Arrow*

1. *Thuật toán*
   1. *Đọc input*
   2. *Dựng đồ hoạ*
2. *Thiết kế giao diện*

CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. *Kết luận*
2. *Hướng phát triển*

TÀI LIỆU THAM KHẢO