1. Introduce
2. Machine learning
3. Lí do chọn đề tài

* Hoàn cảnh:
* Y khoa: Theo báo cáo của Bộ Y tế năm 2017, các bệnh viện lớn và nhỏ trên toàn quốc thường xuyên lâm vào tình trạng quá tải trong khâu chuẩn đoán bệnh. Trong khi các công việc này có sự tương đồng, đơn giản và lặp đi lặp lại hàng ngày. Việc này dẫn đến nhu cầu, các bệnh viện cần có một hệ thống công nghệ thay cho các bác sĩ trong việc phân loại và chuẩn đoán các loại bệnh với độ chính xác cao.
* Công nghệ: Machine Learning nói riêng và AI (Artificial Intelligence) nói chung trong tương lai đang phát triển rất nhanh chóng và hỗ trợ, phục vụ con người trong nhiều lĩnh vực như: kinh tế, công nghiệp, đời sống, giáo dục,… và đặt biệt là Y khoa.
* Lý do tại sao nhóm chọn vấn đề về xương người:
* Vấn đề gãy xương rất phổ biến ở nước ta khi mà tình trạng tai nạn giao thông hằng năm đều rất cao.
* Đây là loại bệnh đơn giản nhưng lại tốn nhiều thời gian để chuẩn đoán và độ rủi ro thấp khi áp dụng công nghệ AI vào.
* Nếu hệ thống được cộng đồng chú ý và có độ chính xác cao, chúng ta có thể áp dụng nó vào những lĩnh vực khác trong Y khoa.

1. Ý nghĩa đề tài

* Áp dụng Machine Learning & AI vào trong y khoa sẽ giúp rút ngắn thời gian khám và chuẩn bệnh của các bác sĩ. Điều này sẽ tiết kiệm công sức, thời gian và chi phí rất nhiều.
* Hỗ trợ các bác sĩ trong việc chuẩn đoán bệnh.

1. Goal

* Đáp ứng được yêu cầu đề ra của giáo viên và người hướng dẫn.
* Học hỏi thêm những kiến thức và công nghệ mới.
* Kết quả đầu ra có độ chính xác cao (>70%).
* Áp dụng vào các hệ thống lớn trong tương lai.

1. Objective

Các bước cần thực hiện để hoàn thành đồ án:

1. Đặt vấn đề

* Phân tích vấn đề
* Lựa chọn giải pháp và thuật toán.

1. Chuẩn bị dữ liệu

* Thu thập dữ liệu.
* Phân loại (dán nhãn) dữ liệu.
* Gray scale (xám hóa) và denoise (giảm nhiễu) cho dữ liệu.

1. Tìm hiểu thuật toán

* Tìm hiểu các định nghĩa, nguyên lý cơ bản về thuật toán đã chọn.
* Nghiên cứu sâu và chạy tay thuật toán.
* Chuyển dữ liệu đầu vào thành feature vectors.

1. Training model

* Viết code.
* Dựa vào các feature vectors và thuật toán đã có để tạo thành một model.

1. Testing

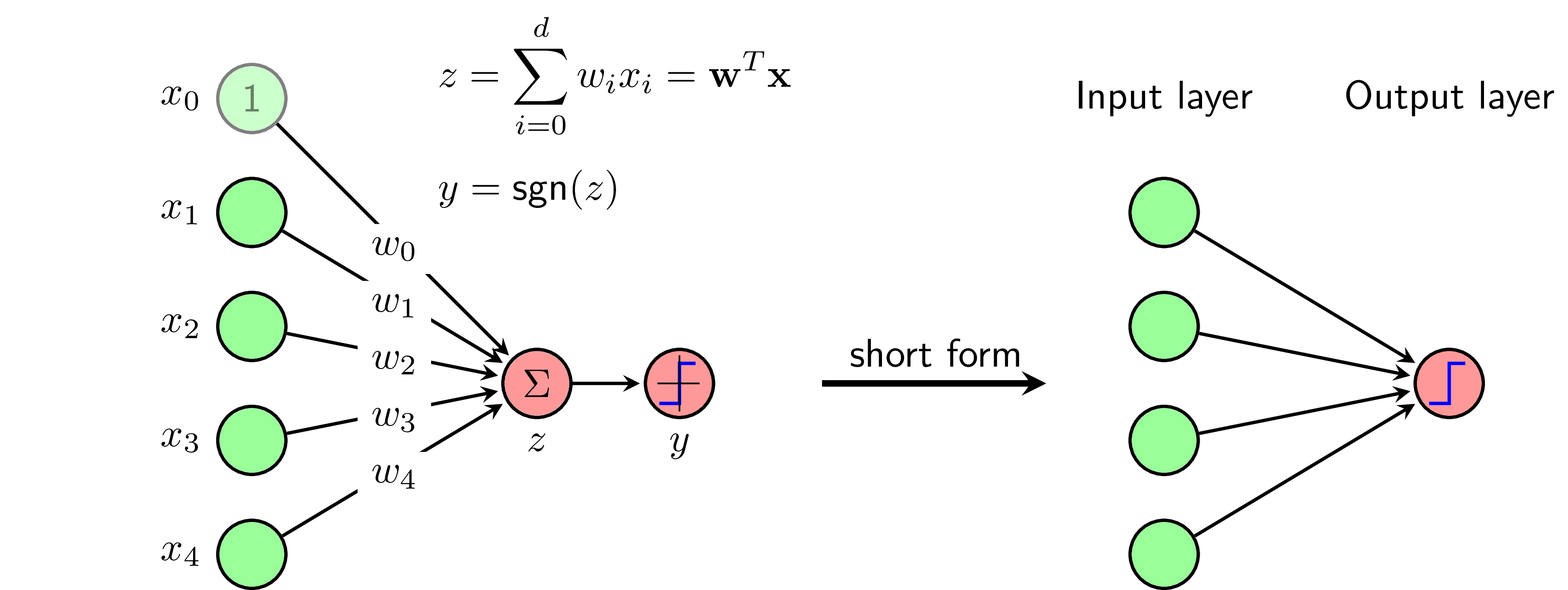
* So sánh dữ liệu test với model đã training.
* Cập nhật lại model nếu kết quả không như mong muốn nhằm tăng độ chính xác của chương trình.

1. Theory

Nhóm mình sẽ sử dụng thuật toán Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập). CNN là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay như hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động. CNN được sử dụng nhiều trong các bài toán nhận dạng các object trong ảnh. CNN chính là một dạnh Artificial Neural Network, một Multiplayer Perceptron nhưng mang thêm 1 vài cải tiến, đó là **Convolution** và **Pooling**. Để sử dụng được CNN, chúng ta cần phải tìm hiểu những kiến thức sau đây:

1. Basic knowledge about Machine Learning
2. Neural Network

Mạng nơron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơron.



Các thành phần cơ bản của một nơron nhân tạo bao gồm:

• Tập các đầu vào: Là các tín hiệu vào (input signals) của nơron, các tín hiệu này thường được đưa vào dưới dạng một vector N chiều.

• Tập các liên kết: Mỗi liên kết đƣợc thể hiện bởi một trọng số liên kết – Synaptic weight. Trọng số liên kết giữa tín hiệu vào thứ j với nơron k thường được kí hiệu là . Thông thường, các trọng số này được khởi tạo một cách ngẫu nhiên ở thời điểm khởi tạo mạng và được cập nhật liên tục trong quá trình học mạng.

• Bộ tổng (Summing function): Thường dùng để tính tổng của tích các đầu vào với trọng số liên kết của nó.

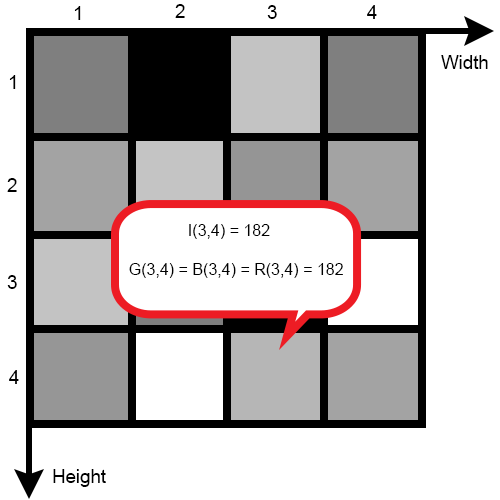
• Ngưỡng (còn gọi là một độ lệch - bias): Ngưỡng này thường được đưa vào như một thành phần của hàm truyền.

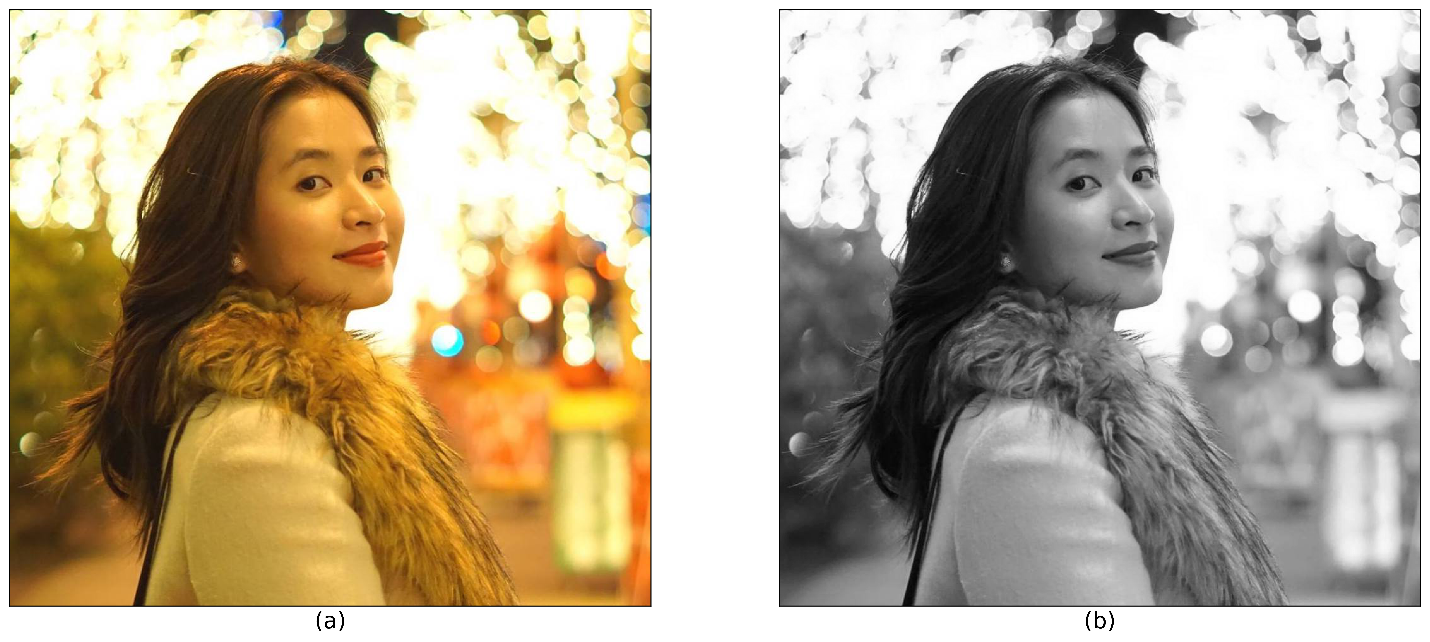
• Hàm truyền (Transfer function): Hàm này được dùng để giới hạn phạm vi đầu ra của mỗi nơron. Nó nhận đầu vào là kết quả của hàm tổng và ngưỡng.

• Đầu ra: Là tín hiệu đầu ra của một nơron, với mỗi nơron sẽ có tối đa là một đầu ra.

1. Gray scale
2. Khái niệm

* Là một hệ thống màu có mô hình màu đơn giản nhất với 256 cấp độ xám biến thiên từ màu đen đến màu trắng.
* Sản phẩm được xuất ra sẽ có màu trắng đen.
* Được sử dụng cả trong công nghiệp in lẫn dùng trong việc thể hiện ảnh lên các thiết bị số.
* Ảnh xám (Gray image) hay còn gọi là ảnh đơn sắc (Monochromatic), mỗi giá trị điểm ảnh (Pixel) trong ma trận điểm ảnh mang giá trị từ 0 đến 255.
* Trong khôn gian màu RGB, để có một ảnh xám cần có phải có giá trị kênh màu Red(x, y) = Green(x, y) = Blue(x, y). (Với x,y lần lượt là tọa độ của điểm ảnh).



1. Chuyển đổi hệ thống màu RGB sang Grayscale

Input Output

1. Convolution layer

Là một layer đầu tiên để tách features từ một bức hình đầu vào. Tích chập bảo toàn mối quan hệ giữa các điểm ảnh bằng cách học features của bức hình sử dụng những hình vuông nhỏ của dữ liệu đầu vào. Nó là một thuật toán mà xem 2 dữ liệu đầu vào là ma trận hình ảnh và một kernel.

Một số khái niệm cơ bản: **Filter**, **Kernel** hay **Feature Detector** đều là cách gọi của ma trận lọc (như mình đã đề cập ở trên). Thông thường, ở các lớp đầu tiên của Conv Layer sẽ có kích thước là [5x5x3]:

* Convolved Feature, Activation Map hay Feature Map là đầu ra của ảnh khi cho bộ lọc chạy hết bức ảnh với phép tích vô hướng.
* Receptive field là vùng ảnh được chọn để tính tích chập, hay bằng đúng cái kích thước của bộ lọc.
* Depth là số lượng bộ lọc. Lưu ý: ở đây là số lượng bộ lọc (filter) chứ không phải số lượng kênh màu RBG như ở trên.
* Stride được hiểu là khoảng cách dịch chuyển của bộ lọc sau mỗi lần tính. Ví dụ khi stride=2. Tức sau khi tính xong tại 1 vùng ảnh, nó sẽ dịch sang phải 2 pixel. Tương tự với việc dịch xuống dưới.
* Zero-Padding là việc thêm các giá trị 0 ở xung quanh biên ảnh, để đảm bảo phép tích chập được thực hiện đủ trên toàn ảnh.

Cách tính kích thước đầu ra của ảnh mỗi layer:

Giả sử ảnh đầu ra là [ x x ]

Thì:

=

=

=

Trong đó:

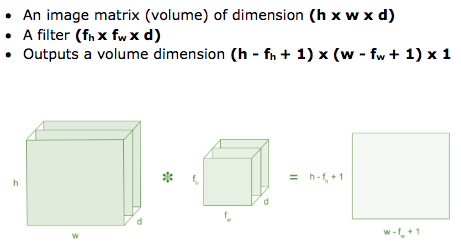
[ x x ]: Kích thước đầu vào

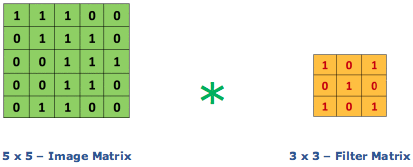
: Kích thước bộ lọc Kernel

: giá trị Stride

: số lượng zero-padding thêm vào viền ảnh

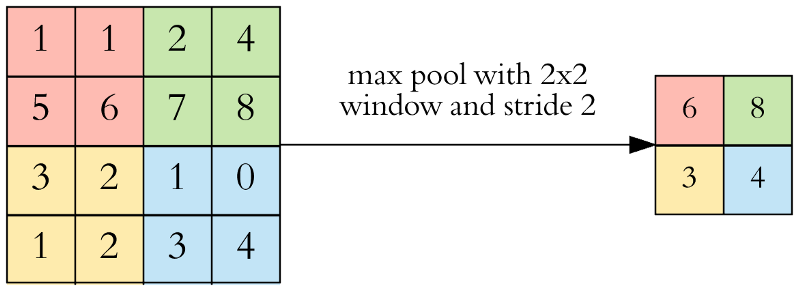
: Số lượng bộ lọc (Depth)





1. Max pooling

Mục đích của pooling rất đơn giản, nó làm giảm số hyperparameter mà ta cần phải tính toán, từ đó giảm thời gian tính toán, tránh overfitting. Loại pooling ta thường gặp nhất là max pooling, lấy giá trị lớn nhất trong một pooling window.



1. ReLU Layer

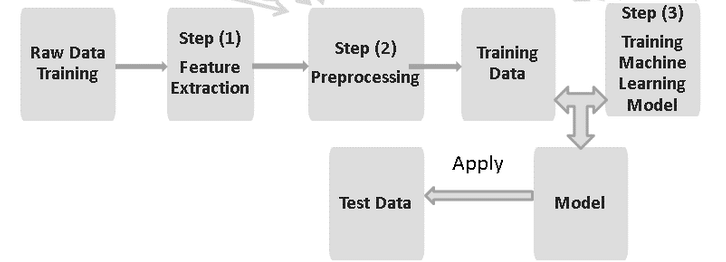
ReLU layer áp dụng các kích hoạt (activation function) **max(0,x)** lên đầu ra của Conv Layer, có tác dụng đưa các giá trị âm về thành 0. Layer này không thay đổi kích thước của ảnh và không có thêm bất kì tham số nào.

Mục đích của lớp ReLU là đưa ảnh một mức ngưỡng, ở đây là 0. Để loại bỏ các giá trị âm không cần thiết mà có thể sẽ ảnh hưởng cho việc tính toán ở các layer sau đó.

1. Fully connected layer

Đang trong quá trình tìm hiểu...

1. Model.

Block diagram

Flow chart