# TÓM TẮT

**TÊN ĐỀ TÀI: NHẬN DẠNG HÀNH VI CON NGƯỜI ỨNG DỤNG VÀO LUYỆN TẬP THỂ DỤC THỂ THAO (GYM ASSISTANT)**

Sinh viên thực hiện: Đặng Tiểu Minh

106180095

Trương Văn Phước

106180105

Lớp: 18DT2

Pose Estimation (đôi khi được dùng với thuật ngữ Skeleton Detection) là một vấn đề khá phổ biến trong lĩnh vực xử lý ảnh khi chúng ta cần xác định vị trí và tư thế của một đối tượng. Mức ý nghĩa ở đây là chúng ta phải rút ra được những đặc điểm chính, những đặc điểm đó là những đặc trưng của đối tượng (có thể mô tả được đối tượng).

Ở trong đề tài này, chúng em muốn đề cập đến bài toán Human activity recognition sử dụng Mediapipe pose và LSTM model, tức là nhận dạng hành động của con người nhờ vào một phần/ toàn bộ các phần chính của cơ thể con người (vd vai, khuỷu tay, cổ tay, đầu gối v.v).

Ở đây chúng em trình bày một cách tiếp cận để phát hiện hành vi, hành động của một người. Cách tiếp cận sử dụng một biểu diễn không theo tỷ lệ để tìm hiểu cách liên kết các bộ phận cơ thể với các cá nhân trong hình ảnh.

Nhận dạng hành động của con người vấn đề kết nối các điểm chính hoặc các bộ phận, trong đó phần lớn tập trung vào việc tìm kiếm chuyển động của các bộ phận cơ thể của cá nhân. Từ đó, ta suy ra hành động hành vi đang thực hiện ở chế độ thời gian thực, bên cạnh đó tích hợp một số chức năng nhận dạng trạng thái đếm số lần lặp lại hành động.

Trong đề tài này chúng em sử dụng thuật toán nhận dạng hành động, hành vi con người, đã được thực hiện bằng cách sử dụng Tensorflow. Nó cũng cung cấp một số biến thể có một số thay đổi về cấu trúc mạng để xử lý thời gian thực trên CPU. Ở đây chúng em sẽ sử dụng mô hình được huấn luyện rồi để chỉ ra các phần chính của cơ thể con người.

Tuy gặp một số khó khăn tuy nhiên vẫn thu được kết quả cơ bản của phần nhận diện này sẽ là dạng tư thế của người sử dụng được nhận trực tiếp từ video với thời gian thực soft real time, bao gồm các điểm chính của cơ thể.

**MỞ ĐẦU**

**Tổng quan**

Mục tiêu của đề tài là xây dựng một hệ thống cho phép dùng camera ghi lại video và dùng các giải thuật nhận dạng khung xương người, qua đó xác định hành động con người dựa trên các phân lớp được định nghĩa sẵn. Kết quả thu được là cơ sở thực hiện các nghiên cứu tiếp theo về dáng đi người hỗ trợ cho các hệ thống như:

Hỗ trợ sinh hoạt: Robot chăm sóc cá nhân tại nhà, bệnh viện có thể được triển khai trong tương lai. Đối với những robot này, việc phát hiện và ước tính tư thế con người có độ chính xác cao là cần thiết để thực hiện nhiều nhiệm vụ khác nhau, chẳng hạn như phát hiện ngã.

Nhân vật hoạt hình: Theo truyền thống, việc tạo hình nhân vật hoạt hình là một quá trình thủ công. Tuy nhiên, các tư thế có thể được đồng bộ hóa trực tiếp với một diễn viên ngoài đời thực thông qua các hệ thống ước tính tư thế chuyên biệt. Các hệ thống cũ dựa vào các điểm đánh dấu hoặc bộ quần áo chuyên dụng. Những tiến bộ gần đây trong ước tính tư thế và chụp chuyển động đã cho phép các ứng dụng trong thời gian thực không cần đánh dấu.

Hệ thống hỗ trợ lái xe thông minh: Tai nạn xe hơi chiếm khoảng 2% số người chết trên toàn cầu mỗi năm. Do đó, hệ thống thông minh theo dõi tư thế tài xế có thể dùng cho các cảnh báo khẩn cấp. Đồng thời, các thuật toán phát hiện người đi bộ đã được sử dụng thành công trên những chiếc xe tự hành, cho phép chiếc xe đưa ra quyết định thông minh hơn.

Ứng dụng y tế: Theo dõi khung xương đã được sử dụng để phát hiện các vấn đề về tư thế như vẹo cột sống bằng cách phân tích các bất thường ở tư thế của bệnh nhân, vật lý trị liệu và nghiên cứu về sự phát triển não bộ nhận thức của trẻ nhỏ bằng cách theo dõi chức năng vận động.

**Mục tiêu của đề tài**

Nghiên cứu tạo ra một ứng dụng có thể giám sát hành vi con người sử dụng thiết bị camera kết nối trực tiếp với Laptop. Trong dự án này sẽ tập trung vào các hành động tập luyện thể dục thể thao như: Pullups (Hít xà), Pushups (Hít đất), Squats (tập chân), Hands (tập cẳng tay). Sự chuyển động của các bài tập được ghi lại bằng camera.

Nghiên cứu thuật toán giúp phân lớp gán nhãn hành động con người dựa vào những dữ liệu đã huấn luyện trước đó.

Xây dựng ứng dụng tích hợp phát hiện trạng thái, số lượng và thời gian hỗ trợ việc tập luyện thể dục thể thao hiệu quả. Đánh giá được hiệu quả của mô hình.

**Nhiệm vụ của đề tài**

Nội dung 1: Nghiên cứu các đặc trưng hành động hành vi, phân tích hành động.

Nội dung 2: Sử dụng camera (webcam) thu thập hành động, hành vi bằng thư viện Mediapipe pose một cách chính xác nhất.

Nội dung 3: Xây dựng mô hình mạng Neural network, train dữ liệu đã thu được từ webcam ở trên.

Nội dưng 4: Viết chương trinh ứng dụng máy tính hiện kết quả của hệ thống bên cạnh đó tích hợp một số chức năng hỗ trợ.

**Đối tượng phạm vi đề tài**

* Các tập dữ liệu dáng đi, tư thế của một hoặc nhiều người.
* Các mô hình mạng Neural và deep learning
* Các thư viện, framework hỗ trợ xử lý ảnh, deep learning với python: OpenCV, TensorFlow, Keras, Mediapipe Pose, Caffe …

**Phương pháp nghiên cứu**

* Nghiên cứu các bài báo liên quan
* Tìm hiểu các phương pháp xử lý ảnh, mô hình hóa.
* Nghiên cứu deep learning và các mô hình neural network.
* Xử lý tập dữ liệu huấn liệu.
* Xây dựng, thử nghiệm các thuật toán, viết chương trình hoàn chỉnh.
* Thực nghiệm với đầu vào là webcam.
* Kiểm tra, tối ưu chương trình, đánh giá kết quả.

**CHƯƠNG I: TỔNG QUAN VỀ DEEP LEARNING VÀ BÀI TOÁN NHẬN DẠNG HÀNH VI, HÀNH ĐỘNG CON NGƯỜI (HUMAN ACTIVITY RECOGNITION)**

**Giới thiệu chương**

Chương một tập trung vào giới thiệt tổng quan về deep learning ứng dụng của nó trong các lĩnh vực đời sống.

Giới thiệu về nhận dạng hành động phổ biến quả các tài liệu nghiên cứu trước đó.

**Tổng quan về Deep learning**

**Deep learning là gì**

Deep Learning là một chủ đề Trí tuệ nhân tạo (AI) và là một phạm trù nhỏ  
của máy học. Deep Learning tập trung giải quyết các vấn đề liên quan đến mạng  
Neural nhân tạo nhằm nâng cấp các công nghệ như nhận diện giọng nói, thị giác máy  
tính và xử lý ngôn ngữ tự nhiên. Deep Learning đang trở thành một trong những lĩnh vực quan trọng trong khoa học máy tính. Chỉ trong vài năm, Deep Learning đã thúc đẩy tiến bộ trong đa dạng các lĩnh vực như nhận thức sự vật (object perception), dịch tự động (machine translation), nhận diện giọng nói - đó là những vấn đề từng rất khó khăn với các nhà nghiên cứu trí tuệ nhân tạo.

**Mạng Neural nhân tạo**

Trong công nghệ thông tin, mạng Neural nhân tạo là một hệ thống các chương  
trình và cấu trúc dữ liệu mô phỏng cách vận hành của não người. Một mạng Neural  
như vậy thường bao gồm một lượng lớn các vi xử lý hoạt động song song, mỗi vi  
xử lý chứa đựng một vùng kiến thức riêng và có thể truy cập vào các dữ liệu trong  
bộ nhớ riêng của mình. Trong mạng Neural nhân tạo, các Neural chính là các node (node là đơn vị Neural trong mạng Neural nhân tạo – mỗi chiếc máy tính trong mạng Neural có thể được xem như 1 node) được kết nối với nhau trong một mạng lưới lớn. Bản thân  
từng node này chỉ trả lời được những câu hỏi hết sức cơ bản chứ không hề thông  
minh, nhưng khi được gộp chung với nhau thì chúng lại có sức mạnh xử lý được cả  
những tác vụ khó. Và điều quan trọng ở đây là bằng những thuật toán phù hợp,  
chúng ta có thể dạy và huấn luyện được chúng.

**Một số ứng dụng của Deep learning**

* Hệ thống gợi ý trên các nền tảng
* Nhận diện hình ảnh
* Xử lý ngôn ngữ tự nhiên

**Bài toán nhận dạng hành động trong thị giác máy tính**

**Tổng quan thị giác máy tính**

Thị giác máy tính (Computer Vision) là một lĩnh vực đã và đang rất phát  
triển. Khái niệm thị giác máy tính có liên quan tới nhiều ngành học và hướng  
nghiên cứu khác nhau. Từ những năm 1970 khi mà năng lực tính toán của máy tính  
ngày càng trở nên mạnh mẽ hơn, các máy tính lúc này có thể xử lý được những tập  
dữ liệu lớn như các hình ảnh, các đoạn phim thì khái niệm và kỹ thuật về thị giác  
máy tính ngày càng được nhắc đến và nghiên cứu nhiều hơn cho tới ngày nay. Hiện  
tại lĩnh vực được các chuyên gia đánh giá là vẫn còn “non nớt” và có rất nhiều sự  
thay đổi trong thời gian tới.

**Thế nào là thị giác máy tính**

Thị giác máy tính bao gồm lý thuyết và các kỹ thuật liên quan nhằm mục đích tạo ra một hệ thống nhân tạo có thể tiếp nhận thông tin từ các hình ảnh thu được hoặc các tập dữ liệu đa chiều. Đối với mỗi con người chúng ta, quá trình nhận thức thế giới bên ngoài là một điều dễ dàng. Quá trình nhận thức đó được “học” thông qua quá trình sống của mỗi người. Tuy nhiên với các vật vô tri vô giác như các máy tính, robot… thì điều đó quả thực là một bước tiến rất gian nan. Các thiết bị ngày nay không chỉ nhận thông tin ở dạng tín hiệu đơn lẻ mà nay còn có thể có cái “nhìn” thật với thế giới bên ngoài. Cái “nhìn” này qua quá trình phân tích, kết hợp với các mô hình như máy học, mạng Neural… sẽ giúp cho thiết bị tiến dần tới một hệ thống nhân tạo có khả năng ra quyết định linh hoạt và đúng đắn hơn rất  
nhiều.

**Ứng dụng của thị giác máy tính**

Lĩnh vực nghiên cứu của thị giác máy tính rất rộng, và đặc điểm chung là các bài toán về thị giác máy tính đều không có một đề bài chung và cách giải duy nhất. Mỗi giải pháp giải quyết vấn đều được một kết quả nhất định cho những trường hợp cụ thể.

* Một vài lĩnh vực mà Computer Vision được ứng dụng có thể kể tới như sau:
* Điều khiển tiến trình (ví dụ: trong các robot công nghiệp, hay các thiết bị, xe tự hành)  
  Phát hiện sự thay đổi (ví dụ: các thiết bị giám sát)
* Tổ chức thông tin (ví dụ: chỉ số kho dữ liệu các ảnh hoặc chuỗi ảnh liên tục) Mô hình hoá đối tượng (ví dụ: quá trình kiểm tra trong môi trường công nghiệp, xử lý ảnh trong y học)
* Tương tác (đóng vai trò làm đầu vào cho thiết bị trong quá trình tương tác giữa người và máy)

**Nhận dạng**

**Giới thiệu**

Nhận dạng ảnh là giai đoạn cuối cùng của các hệ thống xử lý ảnh**.** Nhận dạng là quá trình phân loại các đối tượng được biểu diễn theo một mô hình nào đó và gán cho chúng vào một lớp dựa theo những quy luật và các mẫu chuẩn. Quá trình học dựa vào các mẫu học biết trước gọi là nhận dạng có thầy hay học có thầy (supervised learning). Trong trường hợp ngược lại gọi là học không có thầy (no supervised learning).

Trong lý thuyết nhận dạng, có ba cách tiếp cận khác nhau:

* Nhận dạng dựa vào phân hoạch không gian.
* Nhận dạng cấu trúc.
* Nhận dạng dựa vào kỹ thuật mạng Neural

Hai cách tiếp cận đầu là kỹ thuật kinh điển. Các đối tượng ảnh quan sát và thu được phải trải qua giai đoạn tiền xử lý nhằm tăng cường chất lượng ảnh, làm nổi các chi tiết, trích chọn và biểu diễn các đặc trưng, và cuối cùng là nhận dạng.

Cách tiếp cận thứ ba hoàn toàn khác, dựa vào cơ chế đoán nhận, lưu trữ và phân biệt đối tượng mô phỏng theo hoạt động của hệ thần kinh con người. Do cơ chế đặc biệt, các đối tượng thu nhận bởi thị giác người không cần qua giai đoạn cải thiện mà chuyển ngay sang giai đoạn tổng hợp, đối sánh với các mẫu đã lưu trữ để nhận dạng.

**Bản chất của quá trình nhận dạng**

Quá trình nhận dạng gồm 3 giai đoạn chính:

* Lựa chọn mô hình biểu diễn đối tượng.
* Lựa chọn luật ra quyết định (phương pháp nhận dạng) và suy diễn quá trình học.
* Học nhận dạng.

Khi mô hình biểu diễn đối tượng đã được xác định, quá trình nhận dạng chuyển sang giai đoạn học. Học là giai đoạn rất quan trọng. Thao tác học nhằm cải thiện, điều chỉnh việc phân hoạch tập đối tượng thành các lớp.

Việc nhận dạng chính là tìm ra quy luật và các thuật toán để có thể gán đối tượng vào một lớp.

* Học có giám sát (supervised learning): Kỹ thuật phân loại nhờ kiến thức biết trước gọi là học có giám sát. Đặc điểm cơ bản của kỹ thuật này là người ta có một thư viện các mẫu chuẩn. Mẫu cần nhận dạng sẽ được đem sánh với mẫu chuẩn để xem nó thuộc loại nào.
* Học không có giám sát (unsupervised learning): Kỹ thuật học này phải tự định ra các lớp khác nhau và xác định các tham số đặc trưng cho từng lớp. Do số lớp và những đặc trưng của các lớp không được biết trước, nên kỹ thuật này khó khăn hơn. Kỹ thuật này nhằm tiến hành mọi cách gộp nhóm có thể và chọn lựa cách tốt nhất. Bắt đầu từ tập dữ liệu, nhiều thủ tục xử lý khác nhau nhằm phân lớp và nâng cấp dần để đạt được một phương án phân loại.

**Nhận dạng hành vi, hành động của con người**

Nhận diện hành động của con người là một lĩnh vực nghiên cứu rộng rãi trong thị giác máy tính. Các ứng dụng của nó bao gồm các hệ thống giám sát, phân tích video, robot và một loạt các hệ thống có liên quan đến sự tương tác giữa người và các thiết bị điện tử như giao diện người-máy tính. Nhận diện hành động của con người được phát triển bắt đầu vào đầu năm 1980. Hiện nay, các nghiên cứu chủ yếu tập trung vào việc học và nhận biết các hành động từ chuỗi video. Một hành động là một chuỗi các chuyển động cơ thể con người, và có thể bao gồm nhiều bộ phận cơ thể đồng thời. Từ quan điểm của thị giác máy tính, việc nhận dạng hành động này là để phù hợp với các quan sát (ví dụ: video) với các mẫu được xác định trước đó và sau đó gán cho nó một nhãn là loại hành động. Tùy thuộc vào độ phức tạp, hoạt động của con người có thể được phân loại thành bốn cấp độ:

Cử chỉ, hành động, tương tác và hoạt động của nhóm, và nhiều nghiên cứu theo hướng một cấu trúc từ dưới lên về nhận dạng hoạt động của con người. Các phần chính của hệ thống như vậy bao gồm trích xuất đặc trưng, học tập hành động, phân loại, nhận dạng hành động và phân đoạn. Một quy trình đơn giản gồm ba bước, cụ thể là phát hiện của con người hoặc các bộ phận cơ thể, theo dõi, và sau đó nhận bằng cách sử dụng kết quả theo dõi. Ví dụ, để nhận ra hành động “bắt tay”, cánh tay và bàn tay của hai người được phát hiện trước tiên và theo dõi để tạo ra một mô tả không gian-thời gian của chuyển động của họ. Mô tả này được so sánh với các mẫu hiện có trong dữ liệu huấn luyện để xác định loại hành động. Mô hình này dựa rất nhiều vào tính chính xác của việc theo dõi, điều này không đáng tin cậy trong  
những cảnh lộn xộn. Nhiều phương pháp đã được đề xuất, và có thể được phân loại theo nhiều tiêu chí khác nhau như trong bài luận điều tra hiện có.[1] Poppe thảo luận nhận dạng hành động của con người từ biểu diễn hình ảnh và phân loại hành động riêng rẽ. Weinland khảo sát các phương pháp cho biểu diễn hành động, phân đoạn và nhận dạng. Turaga chia vấn đề nhận dạng thành hành động và hoạt động theo độ phức tạp, và các hướng tiếp cận phân loại theo khả năng của mình để xử lý các mức độ phức tạp khác nhau. Có nhiều tiêu chí phân loại khác nhau. Trong số đó, [3]Aggarwal và Ryoo là một trong những tổng kết toàn diện mới nhất và so sánh của sự tiến bộ quan trọng nhất trong lĩnh vực này. Dựa vào hành động được nhận dạng từ hình ảnh đầu vào trực tiếp, Aggarwal và Ryoo phân chia các phương pháp nhận dạng thành hai loại chính: phương pháp tiếp cận đơn lớp và phương pháp tiếp cận phân cấp. Cả hai đều là thêm loại con phụ thuộc vào các phương pháp biểu diễn đặc trưng và học

**Kết luận chương**

Chương này đã trình bày tổng quan về các vấn đề như thị giác máy tính, nhận diện hành động của con người và Deep learning.

**CHƯƠNG II: MÔ HÌNH HỌC SÂU, CƠ SỞ LÝ THUYẾT VÀ BÀI TOÁN NHẬN DẠNG HÀNH VI, HÀNH ĐỘNG CON NGƯỜI.**

**Giới thiệu chương**

Chương này tập trung tìm hiểu các vấn đề quan trọng liên quan đến bài toán nhận dạng hành vi, hành động con người, lý thuyết cơ bản về mạng Neural và tìm hiểu tổng quan các phương pháp nhận dạng hành vi, hành động con người cũng như một vài kiến trúc phổ biến mạng Neural trong nhận dạng hành vi, hành động con người.

**Lý thuyết về mạng Neural nhận tạo**

Khái niệm này được bắt đầu vào cuối thập kỷ 1800 khi các nhà nghiên cứu cố gắng mô tả hoạt động của trí tuệ con người. Ý tưởng này bắt đầu được áp dụng cho các mô hình tính toán từ mạng Perceptron

**Mạng Neural nhân tạo**

Mạng Neural nhân tạo (Artificial Neural Network – ANN) hay thường được gọi ngắn gọn là mạng Neural là một mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh của sinh vật, bao gồm số lượng lớn các Neural được gắn kết để xử lý thông tin. ANN được giới thiệu năm 1943 bởi nhà thần kinh học Warren McCulloch và nhà logic học Walter Pits, nó hoạt động giống như bộ não của con người, được học bởi kinh nghiệm (thông qua việc huấn luyện), có khả năng lưu giữ các tri thức và sử dụng các tri thức đó trong việc dự đoán các dữ liệu chưa biết (unseen data). Processing Elements (PE) của ANN gọi là Neural, nhận các dữ liệu vào (inputs) xử lý chúng và cho ra một kết quả (output) duy nhất. Kết quả xử lý của một Neural có thể làm input cho các Neural khác.

**Chart, radar chart

Description automatically generated**

Kiến trúc Artificial Neural Network

Kiến trúc chung của một ANN gồm 3 thành phần đó là input layer, hidden  
layer và output layer. Trong đó, lớp ẩn (hidden layer) gồm các Neural, nhận dữ liệu input từ các Neural ở lớp trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Quá trình xử lý thông tin của một ANN như sau:

Diagram

Description automatically generated

Mô hình cấu tạo một Neural

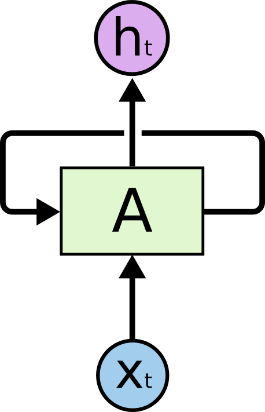
Trong đó, mỗi input tương ứng với 1 thuộc tính của dữ liệu. Ví dụ như trong ứng dụng của ngân hàng xem xét có chấp nhận cho khách hàng vay tiền hay không thì mỗi input là một thuộc tính của khách hàng như thu nhập, nghề nghiệp, tuổi, số con…Output là một giải pháp cho một vấn đề, ví dụ như với bài toán xem xét chấp nhận cho khách hàng vay tiền hay không thì output là yes - cho vay hoặc no – không cho vay. Trọng số liên kết (Connection Weights) là thành phần rất quan trọng của một ANN, nó thể hiện mức độ quan trọng hay có thể hiểu là độ mạnh của dữ liệu đầu vào đối với quá trình xử lý thông tin, chuyển đổi dữ liệu từ layer này sang layer khác. Quá trình học (Learning Processing) của ANN thực ra là quá trình điều chỉnh các trọng số (Weight) của các input data để có được kết quả mong muốn. Hàm tổng (Summation Function) cho phép tính tổng trọng số của tất cả các input được đưa vào mỗi Neural. Hàm tổng của một Neural đối với n input được tính theo công thức sau:

Kết quả trên cho biết khả năng kích hoạt của Neural đó. Các Neural này có thể sinh ra một output hoặc không trong ANN, hay nói cách khác rằng có thể output của 1 Neural có thể được chuyển đến layer tiếp trong mạng Neural hoặc không là do ảnh hưởng bởi hàm chuyển đổi (Transfer Function). Việc lựa chọn Transfer Function có tác động lớn đến kết quả của ANN. Vì kết quả xử lý tại các Neural là hàm tính tổng nên đôi khi rất lớn, nên transfer function được sử dụng để xử lý output này trước khi chuyển đến layer tiếp theo Hàm chuyển đổi phi tuyến được sử dụng phổ biến trong ANN là sigmoid (logical activation) function.

Kết quả của Sigmoid Function thuộc khoảng [0, 1] nên còn gọi là hàm chuẩn hóa (Normalized Function). Đôi khi thay vì sử dụng hàm chuyển đổi, ta sử dụng giá trị ngưỡng (Threshold value) để kiểm soát các output của các Neural tại một layer nào đó trước khi chuyển các output này đến các layer tiếp theo Nếu output của một Neural nào đó nhỏ hơn Threshold thì nó sẽ không được chuyển đến Layer tiếp theo Ứng dụng thực tế của mạng Neural thường được sử dụng trong các bài toán nhận dạng mẫu như nhận dạng chữ cái quang học (Optical character recognition), nhận dạng chữ viết tay, nhận dạng tiếng nói, nhận dang khuôn mặt.

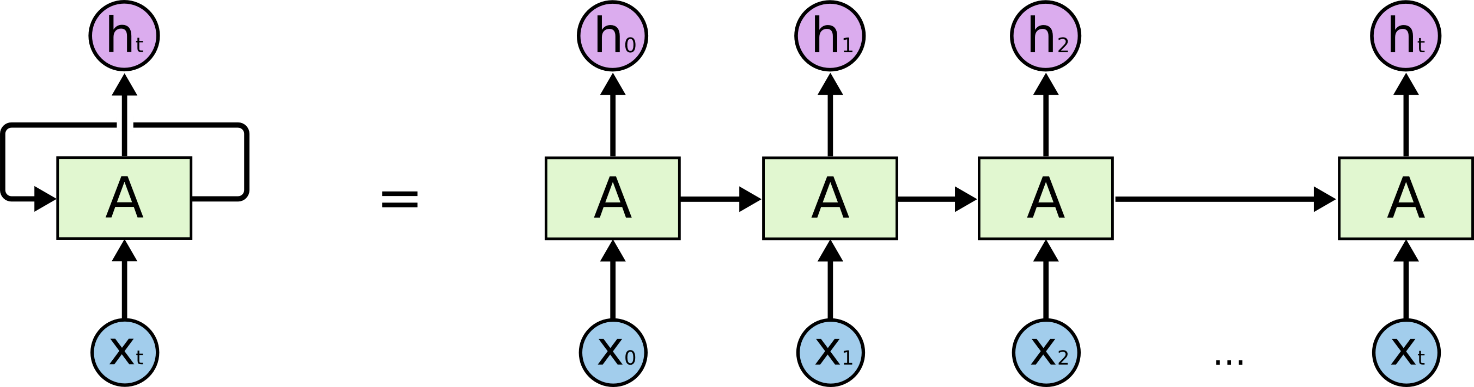
**Mạng Neural hồi quy (RNN - Recurrent Neural Network)**

Mạng Neural hồi quy là trong những mô hình Deep learning được đánh giá có nhiều ưu điểm trong việc xử lý thông tin dạng chuỗi. Recurrent có nghĩa là thực hiện lặp lại cùng một tác vụ cho mỗi thành phần trong chuỗi. Trong đó, kết quả đầu ra tại thời điểm hiện tại phụ thuộc vào kết quả tính toán của các thành phần ở những thời điểm trước đó. Nói cách khác, RNNs là một mô hình có trí nhớ, có khả năng nhớ được thông tin đã tính toán trước đó, không như các mạng Neural truyền thống đó là thông tin đầu vào (input) hoàn toàn độc lập với thông tin đầu ra (output). Về lý thuyết, RNNs có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.



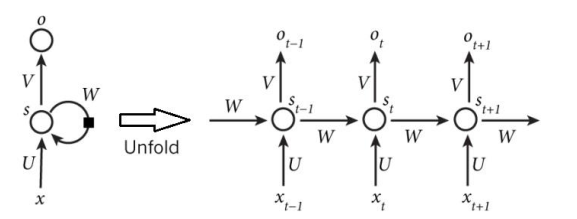
Mô tả mạng Neural hồi quy

Hình vẽ trên mô tả một đoạn của mạng Neural hồi quy A với đầu vào là và đầu ra là  Một vòng lặp cho phép thông tin có thể được truyền từ bước này qua bước này qua bước khác của mạng Neural. Mạng Neural ANN không khác mấy so với mạng Neural thông thuần. Một mạng Neural hồi quy có thể được coi là nhiều bản sao chép của cùng một mạng, trong đó mỗi đầu ra của mạng này là đầu vào của một mạng ANN khác.



Mô tả mạng Neural hồi quy

Huấn luyện RNN tương tự như huấn luyện ANN truyền thống. Giá trị tại mỗi output không chỉ phụ thuộc vào kết quả tính toán của bước hiện tại mà còn phụ thuộc vào kết quả tính toán của các bước trước đó.



Quá trình xử lý thông tin trong mạng RNN

RNN có khả năng biểu diễn mối quan hệ phụ thuộc giữa các thành phần trong chuỗi. Ví dụ, nếu chuỗi đầu vào là một câu có 5 từ thì RNN này sẽ unfold (dàn ra) thành RNN có 5 layer, mỗi layer tương ứng với mỗi từ, chỉ số của các từ được đánh từ 0 tới 4. Trong hình vẽ ở trên, 𝑥𝑡 là input (one-hot vector) tại thời điểm thứ t. 𝑆𝑡 là hidden state (memory) tại thời điểm thứ t, được tính dựa trên các hidden state trước đó kết hợp với input của thời điểm hiện tại với công thức:

𝑆-1 là hidden state được khởi tạo là một vector không. 𝑂𝑡 là output tại thời điểm thứ t, là một vector chứa xác suất của toàn bộ các từ trong từ điển.

Không như ANN truyền thống, tại mỗi layer cần phải sử dụng một tham số  
khác, RNNs chỉ sử dụng một bộ parameters (U, V, W) cho toàn bộ các bước. Trên  
lý thuyết, xây dựng một RNN có thể nhớ được thông tin của một chuỗi dài vô tận  
là có thể, nhưng thực tế xây dựng được như vậy là khó thực hiện vì điều kiện phần  
cứng và thuật toán chưa cho phép.

**Hạn chế mạng RNN**

Về lý thuyết, RNN có thể nhớ được thông tin của chuỗi có chiều dài bất kì, nhưng trong thực tế mô hình này chỉ nhớ được thông tin ở vài bước trước đó.

Hạn chế ở mạng RNN là hidden layer không có trí nhớ dài hạn, hay còn được nhắc tới với tên vanishing gradient problem [7]. Nếu chỉ dừng lại ở việc áp dụng phương án học chuỗi với RNN thì vấn đề độ dài câu đối thoại sẽ là một bài toán khó.

Chart, box and whisker chart

Description automatically generated

RNN phụ thuộc short-term

**Vanishing gradient:** Đạo hàm bị triệt tiêu Là vấn đề xảy ra khi huấn luyện các mạng Neural nhiều lớp. Khi huấn luyện, giá trị đạo hàm là thông tin phản hồi của quá trình lan truyền ngược. Giá trị này trở nên vô cùng nhỏ tại các lớp Neural đầu tiên khiến cho việc cập nhật trọng số mạng không thể xảy ra.

**Long Short Term Memory (LSTM)**

Mô hình này có cấu trúc tương tự như RNNs nhưng có cách tính toán khác đối với các hidden layer. Memory trong LSTMs được gọi là cells (hạt nhân). Ta có thể xem đây là một hộp đen nhận thông tin đầu vào gồm hidden state trước đó và giá trị 𝑥𝑡. Bên trong các hạt nhân này, sẽ quyết định thông tin nào cần lưu lại và thông tin nào cần xóa đi, nhờ vậy mà mô hình này có thể lưu trữ được thông tin dài hạn.

Long Short Term Memory là trường hợp đặc biệt của RNN, có khả năng học long-term dependencies. Mô hình này được giới thiệu bởi Hochreiter & Schmidhuber (1997) [4], và được cải tiến lại. Sau đó, mô hình này dần trở nên phổ biến nhờ vào các công trình nghiên cứu gần đây. Mô hình này có khả năng tương thích với nhiều bài toán nên được sử dụng rộng rãi ở các ngành liên quan. LSTM được thiết kế nhằm loại bỏ vấn đề phụ thuộc quá dài. Ta quan sát lại mô hình RNN bên dưới, các layer đều mắc nối với nhau. Trong RNN chuẩn, module repeating này có cấu trúc rất đơn giản chỉ gồm một lớp đơn giản tanh layer.

LSTM được thiết kế nhằm loại bỏ vấn đề phụ thuộc quá dài. Ta quan sát lại  
mô hình RNN bên dưới, các layer đều mắc nối với nhau thành các mô-đun neural network. Trong RNN chuẩn, mô-đun repeating này có cấu trúc rất đơn giản chỉ  
gồm một lớp đơn giản là tầng tank (**tank layer)**

**Diagram

Description automatically generated**

Các module lặp của mạng RNN chứa một layer

LSTM có các thành phần cơ bản sau:

* Trạng thái tế bào (cell state)
* Cổng (gates)
* Sigmoid
* Tank

Về kiến trúc mạng LSTM**:** giống như RNN, nó là một chuỗi các mô-đun được lặp đi lặp lại. Tuy nhiên, xét về cấu trúc thì LSTM có 4 tầng mạng nơ-ron tương tác với nhau một cách rất đặc biệt, gọi đó là các tầng ẩn (hidden layer). Một số biến thể của LSTM được thực hiện dựa trên việc thay đổi vị trí kết nối giữa các tầng và cổng.

Diagram

Description automatically generated

Các module lặp của mạng LSTM chứa bốn layer

Trong đó, các ký hiệu sử dụng trong mạng LSTM gồm có:

* Hình chữ nhật là các lớp ẩn của mạng Neural.
* Hình tròn biểu diễn toán tử Pointwise.
* Đường kẻ gộp lại với nhau biểu thị phép nối các toán hạng.
* Đường rẽ nhánh biểu thị cho sự sao chép từ vị trí này sang vị trí khác.

Diagram

Description automatically generated

Mô hình thiết kế của LSTM là một bảng mạch số, gồm các mạch logic và các phép toán logic trên đó. Thông tin, hay nói khác hơn là tần số của dòng điện di chuyển trong mạch sẽ được lưu trữ, lan truyền theo cách thiết kế bảng mạch. Mấu chốt của LSTM là cell state (trạng thái nhớ, trạng thái tế bào), đường kẻ ngang chạy dọc ở trên top diagram. Cell state giống như băng chuyền, chạy xuyên thẳng toàn bộ mắc xích, chỉ một vài tương tác nhỏ tuyến tính (minor linear interaction) được thực hiện. Điều này giúp cho thông tin ít bị thay đổi xuyên suốt quá trình lan truyền.

**A picture containing text, clock

Description automatically generated**

Cell state LSTM giống như một băng truyền

LSTM có khả năng thêm hoặc bớt thông tin vào cell state, được quy định một cách cẩn thận bởi các cấu trúc gọi là cổng (gate). Các cổng này là một cách (tuỳ chọn) để định nghĩa thông tin băng qua. Chúng được tạo bởi hàm sigmoid và một toán tử nhân pointwise.

Diagram

Description automatically generated

Cổng trạng thái LSTM

Hàm kích hoạt Sigmoid có giá trị từ 0 – 1, mô tả độ lớn thông tin được phép truyền qua tại mỗi lớp mạng. Nếu ta thu được zero điều này có nghĩa là “không cho bất kỳ cái gì đi qua”, ngược lại nếu thu được giá trị là một thì có nghĩa là “cho phép mọi thứ đi qua”. Một LSTM có ba cổng như vậy để bảo vệ và điều khiển cell state.

**Quá trình hoạt động của mạng LSTM**

Quá trình hoạt động của LSTM được thông qua các bước cơ bản sau.

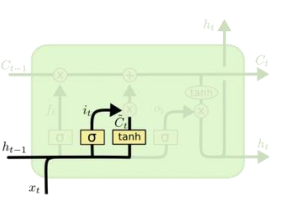
Đầu tiên là sử dụng một tầng sigmoid được gọi là “tầng cổng vào” (input gate layer) để quyết định giá trị nào sẽ cập nhật thông tin nào và chúng ta cần loại bỏ khỏi cell state. Tiến trình này được thực hiện thông qua một sigmoid layer gọi là “forget gate layer” – cổng chặn. Đầu vào là ℎt-1 và 𝑥t, đầu ra là một giá trị nằm trong khoảng [0, 1] cho cell state 𝐶t-1. 1 tương đương với “giữ lại thông tin”, 0 tương đương với “loại bỏ thông tin”.

Diagram

Description automatically generated

*LSTM focus f*

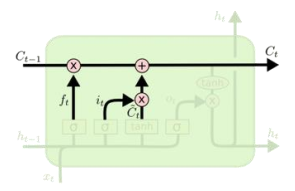
Bước tiếp theo, cần quyết định thông tin nào cần được lưu lại tại cell state. Ta có hai phần là single sigmoid layer được gọi là “input gate layer” quyết định các giá trị chúng ta sẽ cập nhật. Tiếp theo, một tanh layer tạo ra một vector ứng viên mới 𝐶̃𝑡 được thêm vào trong cell state.



*LSTM focus i*

**Cập nhập trạng thái tế bào cũ Ct-1 thành trạng thái mới Ct.**

Ở bước này, sẽ kết hợp hai thành phần này lại để cập nhật vào cell state. Lúc cập nhật vào cell state cũ, Ct-1 vào cell state mới 𝐶t. Ta sẽ đưa state cũ hàm 𝑓t, để quên đi những gì trước đó. Sau đó, ta sẽ thêm 𝑖t ∗ 𝐶̃t. Đây là giá trị ứng viên mới, co giãn (scale) số lượng giá trị mà ta muốn cập nhật cho mỗi state.



*LSTM focus c.*

**Cuối cùng, ta cần quyết định xem đầu ra là gì.**

Cuối cùng, cần quyết định xem thông tin output là gì. Output này cần dựa trên cell state, nhưng sẽ được lọc bớt thông tin. Đầu tiên, áp dụng single sigmoid layer để quyết định xem phần nào của cell state chúng ta dự định sẽ output. Sau đó, ta sẽ đẩy cell state qua tanh (đẩy giá trị vào khoảng -1 và 1) và nhân với một output sigmoid gate, để giữ lại những phần ta muốn output ra ngoài.

A picture containing text, clock

Description automatically generated

*LSTM focus o*

**Kết luận**

LSTM là một cải tiền cho RNN với mục đích giúp cho tất cả các bước của RNN có thể truy vấn được thông tin từ một tập thông tin lớn hơn. LSTM hoạt động thực sự tốt hơn nhiều so với RNN cho nhiều bài toán

Mô hình LSTM là một bước đột phá đạt được từ mô hình RNN

**Hàm sigmoid**

Hàm Sigmoid nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng (0;1). Đầu vào là số thực âm rất nhỏ sẽ cho đầu ra tiệm cận với 0, ngược lại, nếu đầu vào là một số thực dương lớn sẽ cho đầu ra là một số tiệm cận với 1. Trong quá khứ hàm Sigmoid hay được dùng vì có đạo hàm rất đẹp.

Chart, line chart

Description automatically generated

*Đồ thị hàm Sigmoid*

**Hàm Tanh**

Hàm Tanh nhận đầu vào là một số thực và chuyển thành một giá trị trong khoảng

(-1; 1). Cũng như sigmoid, hàm tanh bị bão hoà ở 2 đầu (gradient thay đổi rất ít ở 2 đầu). Tuy nhiên hàm tanh lại đối xứng qua 0 nên khắc phục được một nhược điểm của Sigmoid.

Hàm Tanh còn có thể biểu diễn bằng hàm sigmoid như sau:

Chart, line chart

Description automatically generated

*Đồ thị hàm Tanh*

**Softmax**

Softmax là một cách ràng buộc đầu ra của các mạng nơ ron phải có tổng bằng 1. Qua đó, các giá trị đầu ra của hàm softmax có thể được coi như là một phân phối xác suất của các biến đầu ra. Nó rất hữu ích trong bài toán phân loại đa lớp. Softmax là một loại của hàm kích hoạt - activation function chúng ta đã bàn luận ở phần phía trên với điều đặc biệt đó là kết quả đầu ra của nó sẽ có tổng bằng 1. Để làm được điều này hàm softmax sẽ chuyển đổi giá trị đầu ra của mạng nơ ron bằng cách chia cho tổng giá trị. Lúc này đầu ra có thể coi là một vector của xác suất dự đoán của các class. Chúng ta có thể thấy rõ hơn trong công thức sau:

**Early Stopping**

Khi training model thì không phải lúc nào (hàm mất mát) loss của tập train và tập test cũng đồng thời giảm, tới một epoch nào đó thì loss của tập train sẽ tiếp tục giảm nhưng loss của tập test không giảm mà tăng trở lại => Đó là hiện tượng overfitting. Vì vậy để ngăn chặn nó, thì ngay tại thời điểm đó người ta sẽ dừng việc training ( vì để chương trình tiếp tục training thì cũng không cải thiện được gì mà lại tốn tài nguyên ). Early stopping tức dừng thuật toán trước khi hàm mất mát đạt giá trị quá nhỏ, giúp tránh overfitting.

Vậy dừng khi nào là phù hợp?Một kỹ thuật thường được sử dụng là tách từ training set ra một tập validation set như trên. Sau một (hoặc một số, ví dụ 50) vòng lặp, ta tính cả train error và validation error, đến khi validation error có chiều hướng tăng lên thì dừng lại, và quay lại sử dụng mô hình tương ứng với điểm và validation error đạt giá trị nhỏ.

A picture containing diagram

Description automatically generated

*Early stopping*

**Cross Entropy**

Cross-entropy sử dụng để so sánh khoảng cách giữa các giá trị đầu ra của softmax và one-hot encoding. Cross-entropy là một hàm loss và giá trị của nó có thể được cực tiểu hoá (minimized). Điều này giúp cho một neural networks đánh giá được xác suất (độ chắc chắn) của phép dự đoán một mẫu dữ liệu tương ứng với một class. Xác suất sẽ là lớn nhất đối với biến mục tiêu của chúng ta. Cross entropy là tổng của các xác suất logarit âm. Chúng ta có thể định nghĩa nó theo công thức sau:

L ký hiệu hàm mất mát, là kết quả thực, là kết quả dự đoán

Hàm logarit với giá trị âm được sử dụng để minimize hàm loss (maximize hàm logarit trong công thức tương ứng với việc minimize giá trị âm của nó). Mục đích của việc huấn luyện môt mạng nơ-ron nhân tạo tức là chúng ta sẽ tìm ra được một tập các tham số tối ưu cho bài toán của mình từ các dữ liệu đầu vào. Việc đó tương đương với các thao tác tính toán và cập nhật các trọng số - weights. Công việc tính toán và cập nhật các trọng số đó được gọi là lan truyền ngược - backpropagation. Thủ tục để cực tiểu hóa hàm lỗi - loss function được gọi là tối ưu hóa – optimization.

**Bài toán nhận dạng hành vi, hành động con người**

**Giới thiệu bài toán nhận dạng hành vi, hành động con người**

**Phương pháp nhận dạng phân loại video (Video Classification)**

Nhận diện hành động con người có thể chia làm 2 loại: tĩnh và động. Kiểu tĩnh là dạng hành động như đứng, ngồi, nằm, …, là kiểu hành động không di chuyển trong một khoảng thời gian. Còn kiểu động là mấy hành động như: đi, chạy, nhảy, ... chẳng hạn. Hành động kiểu tĩnh có thể phân loại dễ dàng bằng phân loại ảnh. Trong khi đó kiểu động lại là một chuỗi dịch chuyển của các bộ phận cơ thể người nên dùng phân loại ảnh là không chính xác, mà ta phải xét các frame có chứa hành động đó => phân loại video.

**Các phương pháp phân loại video**

**Single-Frame CNN**

Vẫn là bài toán Image Classification, dùng model CNN predict từng frame. Rồi lấy trung bình cộng các xác suất độc lập của từng frame để ra xác suất cuối cùng.

Đây chính xác là cách tiếp cận sử dụng kỹ thuật của bài toán Image Classification. Chúng ta cho lần lượt từng frame của video chạy qua mô hình Image Classification. Kết quả cuối cùng là trung bình của tất các kết quả dự đoán trên mỗi frame. Tất nhiên, chúng ta có thể chỉ dùng 1 số lượng frames nhất định chứ không cần toàn bộ frames trong video. Giá trị này gọi là window\_size.

**Late Fusion**

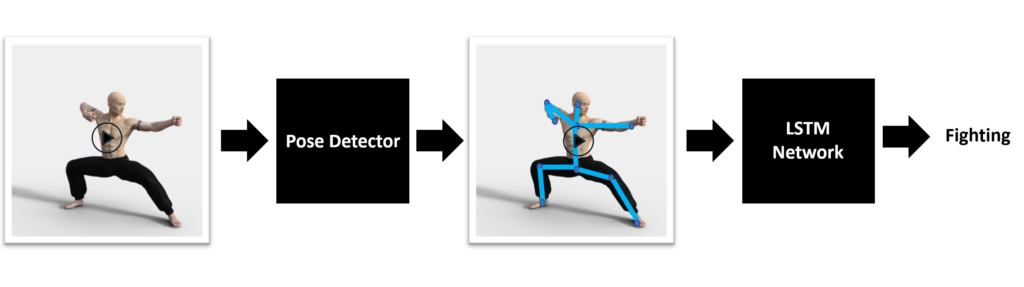
Về cơ bản, phương pháp Late Fusion rất giống với phương pháp Single-Frame CNN nhưng phức tạp hơn một chút. Sự khác biệt ở đây là trong phương pháp Single-Frame CNN, việc tính trung bình trên tất cả các xác suất dự đoán được thực hiện sau khi mô hình đã hoàn thành công việc của nó, nhưng trong phương pháp Late Fusion, quá trình lấy trung bình (hoặc một số kỹ thuật Fusion khác) được tích hợp luôn vào kiến trúc mạng của chính nó. Do đó, cấu trúc thời gian của chuỗi frame được duy trì.

Cụ thể, một Fusion layer được sử dụng để hợp nhất các kết quả đầu ra của của các mạng riêng biệt, những cái hoạt động trên các frames khác nhau. Layer này thường được xây dựng bằng một số kỹ thuật như Max Pooling, Average Pooling, hay Flattening.

Phương pháp Late Fusion được cho là có thể học được các thông tin về không gian và thời gian của những đối tượng, hành động xuất hiện trong video. Do đó, độ chính xác cũng cao hơn phương pháp Single-Frame CNN.

**CNN and Bi-directional LSTM**

**Pose Detection and LSTM**



Mô hình Pose Detection nhận về các Key Points từ cơ thể con người trong mỗi frame. Sau đó, đưa các Key Points này đi qua mạng LSTM đề quyết định hành vi của người đó trong video. Một số mô hình Pose Detection nổi tiếng có thể sử dụng là Open Pose, Detectron2, …Trong bài tiếp theo, chúng ta sẽ sử dụng phương pháp này đề nhận diện các hành vi của con người.

**Kết luận chương, chọn phương pháp phù hợp với bài toán.**

Về phần nội dung của chương 2, chúng ta sẽ hiểu thêm về cơ sở lý thuyết cũng như bài toán nhận dạng về hành vi con người. Ngoài ra có rất nhiều phương pháp hay ở trong chương 2 này nhưng để chọn ra phương pháp phù hợp với bài toán hiện tại là nhận diện vì hành vi thì Pose Detection and LSTM sẽ là phương pháp phù hợp nhất.

**CHƯƠNG III:THIẾT KẾ MÔ HÌNH, CHỨC NĂNG, THỰC HIỆN HỆ THỐNG NHẬN DẠNG**

**Giới thiệu chương**

Trong chương 3 nội dung chủ yếu sẽ là giới thiệu về các thư viện được sử dụng cũng như sơ đồ và lưu đồ thuật toán của hệ thống. Giới thiệu các chức năng của hệ thống nhận dạng, thu thập dữ liệu và cuối cùng là xây dựng mô hình LSTM.

**Giới thiệu về thư viện**

**Python**

Ngôn ngữ Python là một trong những ngôn ngữ lập trình được nhiều người sử dụng nhất hiện nay, nhất là những người mới bắt đầu học lập trình. Không chỉ dễ dàng sử dụng mà  
ngôn ngữ Python còn mang tính hướng đối tượng. Python có cấu trúc dữ liệu cao cấp và hệ thống thư viện lớn nhưng lại có thể tiếp cận đơn giản và vô cùng dễ hiểu. Điểm cộng lớn nhất của ngôn ngữ này chính là sự đơn giản, linh động, và có thể kết hợp với bất kỳ ngôn ngữ lập trình khác, được sử dụng trên nhiều nền tảng và sử dụng để phát triển nhiều ứng dụng.

* Các đặc điểm của Python
* Cú pháp rất tường minh dễ đọc.
* Các khả năng tự xét mạnh mẽ.
* Hướng đối tượng trực giác.
* Các thể hiện tự nhiên mã thủ tục.
* Hoàn toàn mô-dun hóa, hỗ trợ các gói theo cấp bậc.
* Kiểu dữ liệu đông ở mức rất cao.
* Các thư viện chuẩn và các mô-đun ngoài bao quát hầu như mọi việc.
* Phần mở rộng và các mô-đun dễ dàng viếc trong C, C++.
* Có thể nhúng trong ứng dụng như một giao dạng kịch bản (scripting interface).
* Python manh mẽ và thực hiện rất nhanh.

**Tensorflow**

Tensorflow là thư viện mã nguồn mở được phát triển bởi nhóm Google Brain, dành cho tính toán số học và Machine Learning quy mô lớn. Tensorflow kết hợp một loạt các mô hình và thuật toán học máy, học sâu (hay còn gọi là mạng thần kinh) giúp cho quá trình thu thập cơ sở dữ liệu, huấn luyện mô hình, hỗ trợ dự đoán và điều chỉnh kết quả dễ dàng hơn.

Kiến trúc của Tensorflow làm việc trong ba phần:

* Tiền xử lý dữ liệu.
* Dựng mô hình.
* Huấn luyện và ước tính mô hình. Nó được gọi là Tensorflow bởi vì nó nhận đầu vào là một mảng đa chiều, còn được gọi là tensors, các tensor chảy qua một danh sách các toán tử và đi ra ở đầu bên kia.

Các thành phần chính của Tensorflow:

* Variables (Biến): giữ lại giá trị giữa các phiên (session), sử dụng cho trọng số (weights)/ độ lệch (bias). 36
* Nodes (Nút): các phéo tính (operation).
* Tensors: các tín hiệu truyền từ các nút hoặc các tín hiệu truyền đến các nút.  
  Placeholders: được sử dụng để gửi dữ liệu giữa chương trình của người lập trình và biểu đồ tensorflow.

**OpenCV**

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý  
về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có  
tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực.  
OpenCV có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux,  
MacOs lẫn Android, iOS.

* OpenCV có rất nhiều ứng dụng:
* Nhậng dạng hình ảnh
* Xử lý hình ảnh
* Phục hồi hình ảnh / video
* Thực tế ảo

**Keras**

Keras là một API cao cấp được viết bằng Python, chạy trên nền tảng học máy Tensorflow. API cao cấp hơn có nghĩa là Keras có thể đóng vai trò là giao diện người dùng như một front end và Theano (người phát triển chính Yoshua Bengio) hoặc Tensor- flow. Keras được phát triển với trọng tâm là phép thử nghiệm nhanh, với mục tiêu cho phép viết script mà không cần tìm hiểu chi tiết về phần phụ trợ. Keras là một giao diện cao cấp sử dụng Theano hoặc Tensorflow cho phần phụ trợ của nó. Keras hỗ trợ hầu hết các mô hình của mạng neural – fully connected, tích chập (convolutional), gộp (pooling), lặp lại (recurrent), nhúng (embedding), … Các mô hình này có thể được kết hợp để xây dựng mô hình phức tạp hơn.  
Keras hoạt động như một wrapper cho các thư viện cấp thấp. Bản thân Keras sử dụng Tensorflow cho phần back-end. Keras sẽ không hoạt động nếu thực hiện các thay đổi cấp thấp (low-level) cho mô hình. Để có thể thực hiện được thì cần có Tensorflow.

**Sơ đồ hệ thống**

**Quy trình hoạt động**

**Quy trình huấn luyện**

**Diagram

Description automatically generated with medium confidence**

**Quy trình nhận dạng**

**Diagram

Description automatically generated**

**Các chức năng phụ**

**Công thức độ góc skeleton, đếm số Reps, trạng thái.**

Góc của skeleton được lấy từ 3 pose landmarks.

Chức năng đếm số reps sẽ dựa theo chức năng trạng thái đã phân chia các trạng thái ở trên để đếm được người thực hiện đã tập được bao nhiêu lần mỗi bài tập của họ.

Chức năng này sẽ đếm dựa trên góc giữa các điểm như khuỷu tay hay chân để tính là một lần lên hay xuống và nếu đủ một lần lên và xuống thì sẽ được tính là một rep.

Cụ thể là bài tập tay khi người nâng tạ thì khuỷu tay sẽ khép lại một góc nhỏ hơn 50 độ thì sẽ tính là một lần lên còn lúc xuống khuỷu tay lớn hơn 140 độ thì sẽ tính là một lần xuống và đủ một lần lên và xuống thì được tính là hoàn thành một lần tập tay.

Lưu đồ thuật toàn

Ngoài ra hệ thống còn cho phép người dùng có thể lựa chọn các mốc có sẵn như 5, 10, 15 thì sau khi người tập hoàn thành các mốc mà mình đã lựa chọn từ trước đó thì hệ thống sẽ thông báo hoàn thành, mốc free sẽ cho phép người dung tập mà không bị giới hạn.

**Đếm ngược thời gian**

Đây là chức năng giúp người dùng tính thời gian tập luyện. Để người dùng có thể hẹn giờ tập trong bao lâu hoặc có thể đặt ra kế hoạch tập luyện hiệu quả.

Với chức năng này người dùng có thể đặt thời gian mà mình mong muốn, ví dụ như đặt thời gian tập trong 15 phút thì chức năng sẽ đếm ngược từ 15 phút về 0 và hệ thống sẽ thông báo cho người dùng là đã hết thời gian. Và nếu người dùng vẫn muốn tiếp tục tập luyện thì có thể đặt thêm thời gian để tiếp tục tập.

**Tính toán calo**

Chức năng tính toán calo sẽ giúp người dùng biết được số calo mình đã tiêu hao để có thể theo dõi tiến độ tập luyện một cách hiệu quả cũng như tạo động lực giúp người dùng tập luyện tốt hơn.

Bên cạnh đó chức năng còn cho biết số calo/s để người dùng có thể biết được năng suất tập luyện của mình để dễ dàng điều chỉnh hay cải thiện giúp cho bài tập của mình hiệu quả hơn và năng suất tập luyện sẽ được cải thiện hơn rất nhiều.

**Thu thập dữ liệu – Tạo cơ sở dữ liệu**

B1: Dữ liệu của hệ thống sẽ được thu thập dựa trên thời gian thực bằng camera điện thoại thông minh được kết nối với laptop.

B2: Sau đó laptop sẽ đọc ảnh từ camera và dựa vào các điểm Skeleton trên cơ thể con người nhờ vào thư viện MediaPie Pose để phân tích mà phân loại các hành động.

B3: Sử dụng code python để lưu thông số các điểm trên vào một file csv

B4: Tạo dữ liệu để đưa vào model LSTM

B5: Train dữ liệu sau đó lưu model

1. **Xây dựng mô hình mạng LSTM**
   1. **Siêu tham số**
   2. **Thuật toán tối ưu Adam**
   3. **Quá trình huấn luyện**

[1] R. Poppe, “A Survey on Vision-Based Human Action Recognition,” Image and Vision Computing, Vol. 28, No. 6, 2010, pp. 976-990. doi:10.1016/j.imavis.2009.11.014

[2] P. Turaga, R. Chellappa, and A. Veeraraghavan, “Advances in Video-based Human Activity Analysis”, in Advances in Computers, Elsevier, vol. 80, July 2010.

[3] J. K. Aggarwal and M. S. Ryoo, “Human Activity Analysis: A Review,” ACM Computing Surveys, vol. 43 (3), 2011.

[4] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation,Vol.9,No.8,1997, pp. 1735-1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735