**Em xin kính chào thầy cô và các bạn, sau đây em xin trình bày khóa luận tốt nghiệp của mình. Khóa luận của em là XÂY DỰNG ỨNG DỤNG ĐIỀU KHIỂN ĐIỀU HÒA TỰ ĐỘNG DỰA TRÊN NHU CẦU NGƯỜI SỬ DỤNG. Em là nguyễn Thành Long và thầy hướng dẫn em để hoàn thành khóa luận là PGS.TS Nguyễn Hoài Sơn.**

**Sau đây là những nội dung chính em sẽ trình bày. Phần thứ nhất là Phần giới thiệu và đặt vấn đề, Phần thứ hai là trình bày về tổng quan hệ thống. Phần thứ 3 là kết quả thu được sau khi triển khai, và phần cuối cùng là phần tổng kết.**

Đầu tiên, em xin trình bày ngắn gọn về đề tài và mục đích chính của đề tài này. Mục đích của đề tài của em là áp dụng công nghệ IoT cung cấp dịch vụ cho nhà thông minh. Trong phạm vi khóa luận của mình, em sử dụng điều hòa của Daikin để thực nghiệm và đưa ra dịch vụ kiểm soát nhiệt độ phòng dựa vào điều hòa khi người dùng trong trạng thái ngủ.

Như các thầy có thể thấy ở hình sau đây, nó biểu diễn mối quan hệ giữa nhiệt độ cơ thể và giấc ngủ. Theo như những gì mà em tìm hiểu được, giấc ngủ của chúng ta có tính chu kỳ, và chu kỳ này 5 giai đoạn bao gồm ru ngủ, ngủ nông, ngủ sâu, ngủ rất sâu và ngủ mơ.

Và qua mỗi giai đoạn này, nhiệt độ cơ thể của chúng ta lại khác nhau, và thường có xu hướng là giảm dần và sẽ tăng dần khi bắt đầu tỉnh giấc. Nhiệt độ cơ thể sẽ thấp nhất ở giai đoạn ngủ rất sâu. Và một trong những yếu tố ảnh hưởng đến giấc ngủ đó chính là nhiệt độ. Nếu như nhiệt độ trong phòng quá nóng hoặc quá lạnh sẽ làm chúng ta tỉnh giấc, hoặc nghiêm trọng hơn có thể gây cảm hoặc sốt, sốc nhiệt nếu điều chỉnh điều hòa không hợp lý. Chính vì lý do đó, em sẽ đưa ra giải pháp để kiểm soát nhiệt độ phòng tự động khi ngủ để giải quyết vấn đề nêu trên.

Nhiệm vụ chính và những công việc đã làm của em trong đề tài lần này là triển khai và cài đặt IoT platfrom cho nhà thông minh. Sau đó tích hợp điều hòa thông minh vào IoT platform này. Giao thức giao tiếp em sử dụng là Echonet Lite và MQTT. Sau khi đã cài đặt IoT platform và tích hợp điều hòa thành công, em phát triển mobile app để điều khiển điều hòa và đưa ra dịch vụ điều khiển tự động khi người dùng bắt đầu đi ngủ. Chi tiết các phần đã làm em sẽ trình bày ngay sau đây.

(1p)

----

Việc đầu tiên em thực hiện là cài đặt IoT platfrom cho nhà thông minh. Cấu trúc của một IoT platform sẽ gồm những thành phần như sau: Server Side, Local Side và User Side. Về phía server em triển khai MQTT Broker và RESTful API Server, hệ quản trị CSDL MySQL để lưu trữ dữ liệu người dùng và dữ liệu từ các thiết bị; Local Side gồm có điều hòa thông minh + controller và Home Gateway; Phần User side sẽ là mobile APP giúp cho việc tương tác với điều hòa.

Về phần tích hợp điều hòa thông minh vào IoT Platform, điều hòa đã lắp đặt controller hỗ trợ giao tiếp Echonet Lite. Đây là giao thức thường được sử dụng bới các tập đoàn bên Nhật, thích hợp với các thiết bị có dụng lượng bộ nhớ và tốc độ xử lý hạn chế. Home Gateway em đã cài đặt và phát triển để nhận diện, và tìm kiếm các thiết bị Echonet Lite nhờ sử dụng bộ thư viện OpenEcho. Ngoài việc có thể nhận diện được, thì em cũng sử dụng appending echonet lite để giao tiếp với điều hòa …

Về phần dịch vụ và điều khiển tự động nhiệt độ. Ngoài việc có thể điều khiển từ xa, điều khiển các chức năng cơ bản thì em cung cấp them dịch vụ điều khiển tự động khi người dùng ở trong trạng thái ngủ. Ở đây em đã tìm hiểu các yếu tố ảnh hưởng đến giấc ngủ, trong đó nhiệt độ là 1 trong những yếu tố quan trọng ảnh hưởng đến giấc ngủ của chúng ta. Qua các chu kỳ của giấc ngủ, nhiệt độ của cơ thể lại có sự khác biệt, nguyên do là khi chúng ta ngủ, hầu hết các hoạt động sẽ dừng lại, dẫn đến nhiệt độ cơ thể sẽ giảm dần, và trong các chu kỳ ngủ ở đây nhiệt độ cung sẽ giảm dần qua các giai đoạn.

APPENDIX Detailed Requirements for ECHONET Device objects

(1p)

----

Giới thiệu qua mạng Unet: Trong đề tài này, mạng Unet mà em sử dụng là mạng Unet cơ bản với 2 phần encoder bên trái và decoder bên phải. Phần encoder là phần trích đặc trưng em xây dựng với 5 lớp double convolution với kernel size là 3. Giữa các lớp layer là các lớp max pooling 2x2. Như vậy sau mỗi layer, kích thước của ảnh sẽ giảm đi 2 lần và độ sâu của ảnh tăng lên 2 lần. Độ sâu của ảnh ở đây là số lượng bit cần để lưu trữ 1 pixel của ảnh. 1 Pixel của ảnh cần nhiều bit lưu trữ thì tổng thể ảnh càng sắc nét, ít nhiễu.

Tiếp theo là phần decoder, nhiệm vụ của phần decoder là khôi phục lại kích thước của ảnh bởi mỗi lớp layer bên phần encoder đã làm kích thước của ảnh giảm đi 1 nửa để trích xuất đặc trưng. Có một bước đặc biệt trong mạng Unet đó là các phép nối, nối giữa 2 lớp layer đối xứng giữa phần encoder và decoder. Theo như em tìm hiểu thì sau khi thu nhỏ ảnh ở phía encoder, thì ảnh sẽ mất đi mối tương quan về vị trí trong ảnh, thì việc nối 2 ma trận, 2 layer lại với nhau sẽ bổ sung thêm thông tin để cải thiện hình ảnh đầu ra.

(2p)

----

Mạng NASUnet là mạng được tự động tìm kiếm dựa trên xương sống hình chữ U nên gọi là NASUnet. Quá trình để thiết kế ra một kiến trúc mạng thường sẽ rất tốn chi phí và công sức. Vì vậy gần đây tìm kiếm kiến trúc thần kinh được xuất hiện rất nhiều trong các nghiên cứu về Deep learning. Mục đích của NAS để giảm thiểu công việc thủ công của con người cũng như tiết kiệm cho phí cho việc thiết kế ra kiến trúc mới. Mô hình mạng NASUnet em sử dụng trong bài toán này là mạng có sẵn, được tìm kiếm trên tập dữ liệu Pascal VOC là tập dữ liệu hình ảnh về y tế. Công việc chính của em khi làm việc với mạng NAS-Unet này là em add tập dữ liệu địa chấn vào mô hình, và thực hiện xử lý ảnh đầu vào giống với xử lý của mạng Unet để so sánh và nhận xét kết quả. Theo như em tìm hiểu về Nas-Unet thì kiến trúc mạng bao gồm 8 cell (8 ô) nvu chính của 4 ô bên trái cũng gọi là encoder và 4 ô bên phải gọi là decoder. Việc tìm kiếm ra kiến trúc thần kinh bao gồm 3 khía cách: không gian tìm kiếm, chiến lược tìm kiếm, hiệu suất tìm kiếm.

(1p)

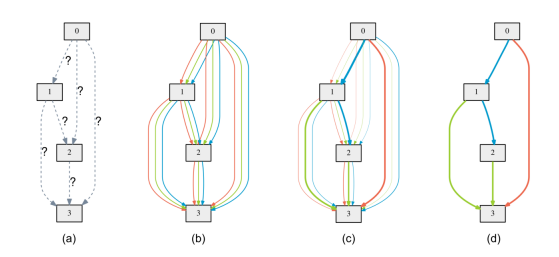
Không gian tìm kiếm

Chiến lược tìm kiếm

Về khía cạnh chiến lược tìm kiếm thì bao gồm các phương pháp chính đó là pp tiến hóa nơ ron, pp học tăng cường và kiến trúc one-hot. Trong mô hình NASUnet mà em sử dụng trong khóa luận lần này người ta sử dụng phương pháp DARTS. Kiến trúc onehot gồm 3 pp là chia sẻ tham số, hypernetwork và tìm kiếm phân biệt. DARTS thuộc trong phần tìm kiếm phân biệt.

DARTS chỉ tối ưu hóa các cấu trúc con được gọi là các ô được xếp chồng lên nhau để xác định kiến trúc mạng đầy đủ. ()()()6p rồi

Đây là hình ảnh mô tả không gian tìm kiếm ở dạng chung, việc tính toán và tìm kiếm kiến trúc này là cho một ô của mô hình được biểu diễn dưới dạng đồ thị xoay chiều có hướng. Ô đã học được có thể được nối đệ quy để tạo thành một mạng lặp lại. Mỗi ô là một đồ thị xoay chiều có hướng gồm N nút có thứ tự



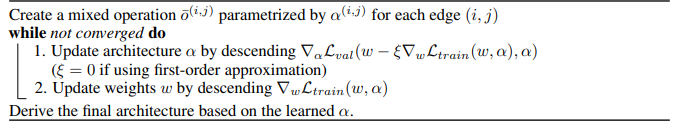
Hình 1: Tổng quan về DARTS: (a) Các hoạt động trên các cạnh ban đầu chưa được biết đến. (b) Các trọng số được cập nhật ở tất cả các phép toán thực hiện (c) Tối ưu hóa chung của xác suất trộn và trọng số mạng bằng cách giải bài toán tối ưu hóa (d) Tạo ra kiến trúc cuối cùng từ các xác suất trộn đã học.

H1

Mỗi nút x(i) sẽ đại diện cho bản đồ tính năng và cạnh có hướng (I, j) đại diện cho một phép toán o(I, j). Giả sử ô có hai nút đầu vào và một nút đầu ra duy nhất. Đối với các ô tích chập, các nút đầu vào được xác định là đầu ra của ô trong hai lớp trước đó. Đối với các ô lặp lại, chúng được định nghĩa là đầu vào ở bước hiện tại và trạng thái được thực hiện từ bước trước đó. Như vậy đầu vào của mỗi nút trung gian được tính toán dựa trên tất cả các nút tiền nhiệm của nó.

X(j) = ∑o(i, j)(x(i))

Nhiệm vụ của tìm kiếm kiến ​​trúc sau đó giảm xuống thành việc tìm hiểu xem phép toán nào được sử dụng nhiều nhất thì việc cập nhật trọng số sẽ diễn ra ở phép toán đó. Sau khi việc cập nhật trọng số hoàn thành DARTS sẽ tối ưu hàm mất mát bằng thuật toán gradient decent. Thuật toán Gradient decent giúp tinh chỉnh các hyperparameter (siêu tham số) đã hợp lý hay chưa.



Đây là mã giả mô tả thuật toán DARTS. Thuật toán sẽ lặp đi lặp lại việc cập nhất kiến trúc dựa vào việc giảm dần của hàm loss validate và cập nhật trọng số. Việc tìm kiếm kiến trúc sẽ dừng lại khi mô hình được hội tụ.

(2p)

Đánh giá hiệu suất

(1p)

----

Kết quả + Nhận xét

Đối với kết quả thử nghiệm mà em thu được của hai mô hình Unet và Nas-Unet. Kết quả Unet nằm trong khoẳng 0.8 0.9 còn NAS-Unet là 0.75 – 0.8. Kết quả trong khoảng này đối với em là hoàn toàn chấp nhận được bởi thời gian đạo tạo cho 1 epoch của NAS-Unet chỉ bằng 1/ 10 cho đến 1/ 7 thời gian đào tạo cho 1 epcoh đối với mạng Unet và tham số sử dụng mạng NAS-Unet chỉ bằng 1/30 so với tham số sử dụng cho mạng Unet. Như vậy khi huấn luyện mô hình với tập dữ liệu lớn hơn, nhiều epoch hơn thì NAS-Unet sẽ tiết kiệm tài nguyên hơn rất nhiều so với Unet. Việc Nas-Unet có kết quả thấp hơn một chút so với Unet có thể là do việc mô hình mạng Unet được tìm kiếm trên tập dữ liệu về y tế, hoặc do vấn đề xử lý ảnh đầu vào của em còn cơ bản.

Công việc trong tương lai:

Thử nghiệm tìm kiếm kiến trúc thần kinh trên chính tập dữ liệu địa chấn

Bổ sung các bước xử lý ảnh đầu vào: Agurement ảnh, …

Cảm ơn thầy cô đã lắng nghe.

(2p)

----