15-18 trang

1. Lý thuyết
2. Tập dữ liệu
   1. Mô tả (mục 2.1)
   2. Cách thức thu thập dữ liệu

Mục tiêu: Mục tiêu của tập dữ liệu này là thu thập dữ liệu cảm biến từ một chiếc quạt ở nhiều trạng thái khác nhau để tạo tập dữ liệu cho các mô hình học máy đào tạo nhằm phát hiện và phân loại chính xác các trạng thái này.

b.1. Thiết lập

Phần cứng: Đảm bảo rằng quạt được trang bị các cảm biến có khả năng thu thập dữ liệu liên quan như độ rung, giá trị dòng điện và điện áp.

Phần mềm: Chuẩn bị môi trường phần mềm thu thập dữ liệu có khả năng giao tiếp với các cảm biến của quạt và lưu trữ dữ liệu được thu thập một cách có tổ chức. Cụ thể ở đây, ta sẽ dùng ngôn ngữ Python để thu thập dữ liệu từ các cảm biến và lưu trong tệp định dạng csv.

b.2. Quy trình thu thập dữ liệu

Khởi tạo: Bắt đầu bằng cách khởi tạo phần mềm thu thập dữ liệu và đảm bảo rằng tất cả các cảm biến đều được hiệu chỉnh và hoạt động đúng cách.

Định nghĩa Nhãn: Xác định sáu nhãn đại diện cho các trạng thái khác nhau của quạt: tắt, bật, chạm, rung lắc, lỗi cánh và gãy cánh.

Giao thức sử dụng: USART.

b.3. Thu thập cho từng trạng thái

|  |  |
| --- | --- |
| Trạng thái | Mô tả |
| Tắt | Mô tả: Quạt đã tắt hoàn toàn.  Thủ tục:   * Đặt quạt ở trạng thái tắt. * Bắt đầu ghi dữ liệu. * Duy trì thời lượng ghi đủ để ghi lại các chỉ số cảm biến cơ bản. * Dừng ghi dữ liệu. |
| Bật | Mô tả: Quạt hoạt động bình thường ở tốc độ tiêu chuẩn.  Thủ tục:   * Bật quạt ở tốc độ hoạt động tiêu chuẩn. * Bắt đầu ghi dữ liệu. * Cho phép quạt chạy trong khoảng thời gian định trước. * Dừng ghi dữ liệu. |
| Chạm | Mô tả: Quạt chịu tác động gõ nhẹ hoặc tác động từ bên ngoài.  Thủ tục:   * Chạm nhẹ vào quạt hoặc để quạt chịu tác động từ bên ngoài có kiểm soát. * Bắt đầu ghi dữ liệu ngay sau khi chạm. * Ghi lại phản hồi của cảm biến của quạt đối với thao tác chạm. * Dừng ghi dữ liệu. |
| Rung lắc | Mô tả: Quạt mô phỏng tình trạng rơi hoặc mất ổn định khi lắp đặt.  Thủ tục:   * Mô phỏng tình huống lắp đặt rơi hoặc mất ổn định của quạt. * Bắt đầu ghi dữ liệu ngay sau cú rơi mô phỏng. * Ghi lại dữ liệu cảm biến trong quá trình rơi và hậu quả của nó. * Dừng ghi dữ liệu. |
| Lỗi cánh | Mô tả: Quạt gặp lỗi ở cánh quạt khi đang hoạt động.  Thủ tục:   * Gây ra lỗi ở một hoặc nhiều cánh quạt (ví dụ: do uốn cong hoặc làm hỏng chúng). * Bắt đầu ghi dữ liệu ngay sau khi xuất hiện lỗi. * Ghi lại dữ liệu cảm biến để phát hiện mọi bất thường hoặc bất thường do lỗi lưỡi cắt. * Dừng ghi dữ liệu. |
| Gãy cánh | Mô tả: Quạt bị gãy hoàn toàn một hoặc nhiều cánh quạt.  Thủ tục:   * Mô phỏng sự gãy hoàn toàn của một hoặc nhiều cánh quạt. * Bắt đầu ghi dữ liệu ngay sau khi cánh quạt bị gãy. * Ghi lại dữ liệu cảm biến để xác định mọi thay đổi mạnh mẽ trong hoạt động của quạt. * Dừng ghi dữ liệu. |

4. Lưu trữ và quản lý dữ liệu

Sắp xếp dữ liệu đã thu thập vào các thư mục hoặc tệp riêng biệt cho từng nhãn để duy trì sự rõ ràng và tạo điều kiện thuận lợi cho việc phân tích trong tương lai.

Đảm bảo rằng mỗi mục nhập dữ liệu đều được đánh dấu thời gian và bao gồm siêu dữ liệu có liên quan như kiểu quạt, thông số kỹ thuật cảm biến và điều kiện môi trường.

5. Kết luận

Sau khi hoàn tất quy trình thu thập dữ liệu cho tất cả các trạng thái, hãy xác minh tính toàn vẹn và chất lượng của tập dữ liệu được thu thập.

Tiến hành mọi quá trình tiền xử lý hoặc lọc dữ liệu cần thiết để loại bỏ nhiễu và đảm bảo tính phù hợp của dữ liệu đó cho việc đào tạo các mô hình học máy.

Tập dữ liệu thu được có thể được sử dụng để phát triển và đánh giá các thuật toán học máy cho các nhiệm vụ phân loại và phát hiện trạng thái quạt.

1. Tiền xử lý dữ liệu
   1. FFT

"Biến đổi Fourier nhanh" (FFT) là một phương pháp đo quan trọng trong khoa học. Nó chuyển đổi tín hiệu thành phổ riêng lẻ và từ đó cung cấp thông tin tần số của tín hiệu. FFT được sử dụng để phân tích lỗi, kiểm soát chất lượng và giám sát tình trạng của máy móc hoặc hệ thống. Bài viết này giải thích cách hoạt động của FFT, các thông số liên quan và ảnh hưởng của chúng đến kết quả đo.

* 1. Kalman

1. Mô hình
   1. Các layers

|  |  |  |
| --- | --- | --- |
| Layer | Properties | Meaning |
| Input | 1044 features | Mảng 1 chiều các đặc trưng |
| Reshape | 1044 columns | Chuyển mảng 1 chiều đặc trưng thành mảng 2 chiều với số cột là số đặc trưng.  (ma trận chuyển vị) |
| 1D Conv / Pool |  | Mạng nơ-ron tích chập. Học các đặc trưng theo 1 chiều duy nhất. |
| 32 neurons | Số lượng bộ lọc được sử dụng (32) |
| 5 kernel size | Kích thước cửa sổ trượt (5x5) |
| 1 layer | Số lượng lớp tích chập (1) |
| Dropout | 0.1 rate | Giảm thiểu nguy cơ model bị overfitting bằng cách ngẫu nhiên loại bỏ 1 phần của các kết nối trong mạng.  Tỉ lệ kết nối bị loại bỏ ngẫu nhiên. |
| 1D Conv / Pool |  | Mạng nơ-ron tích chập. Học các đặc trưng theo 1 chiều duy nhất. |
| 16 neurons | Số lượng bộ lọc được sử dụng (16) |
| 3 kernel size | Kích thước cửa sổ trượt (3x3) |
| 1 layer | Số lượng lớp tích chập (1) |
| Flatten |  | Chuyển dữ liệu đa chiều sang đơn chiều  (Sau khi dữ liệu đi qua mạng tích chập cần “làm phẳng” để trở về trạng thái ban đầu). |
| Dense | 128 neurons | Mạng kết nối đầy đủ gồm 128 nodes |
| Dropout | 0.15 | Giảm thiểu nguy cơ model bị overfitting bằng cách ngẫu nhiên loại bỏ 1 phần của các kết nối trong mạng.  Tỉ lệ kết nối bị loại bỏ ngẫu nhiên. |
| Dense | 64 neurons | Mạng kết nối đầy đủ gồm 64 nodes |

A diagram of a software system

Description automatically generated with medium confidence

* 1. Optimization

Adam là một thuật toán cho kỹ thuật tối ưu hóa cho việc giảm độ dốc. Phương pháp này thực sự hiệu quả khi giải quyết vấn đề lớn liên quan đến nhiều dữ liệu hoặc tham số. Nó đòi hỏi ít bộ nhớ hơn và hiệu quả. Về mặt trực quan, nó là sự kết hợp giữa thuật toán ‘gradient descent with momentum’ và thuật toán 'RMSP'.

Tối ưu hóa Adam bao gồm sự kết hợp của hai phương pháp giảm độ dốc:

* Momentum (quán tính): Thuật toán này được sử dụng để tăng tốc thuật toán giảm độ dốc bằng cách xem xét ‘trung bình có trọng số theo cấp số nhân’ của độ dốc. Việc sử dụng giá trị trung bình làm cho thuật toán hội tụ về cực tiểu với tốc độ nhanh hơn. , với
  + = tổng các gradient tại thời điểm t [hiện tại] (ban đầu, = 0)
  + = tổng các gradient tại thời điểm t-1 [trước]
  + = trọng lượng tại thời điểm t
  + = trọng lượng tại thời điểm t+1
  + = tốc độ học tại thời điểm t
  + = đạo hàm của hàm mất mát
  + = đạo hàm của trọng số tại thời điểm t
  + β = Tham số trung bình động (hằng số, 0.9)
* RMSP: Căn bậc hai prop bình phương hoặc RMSprop là một thuật toán học thích ứng nhằm cố gắng cải thiện AdaGrad. Thay vì lấy tổng tích lũy của các gradient bình phương như trong AdaGrad, nó lấy ‘trung bình di chuyển theo cấp số nhân’.
  + = trọng lượng tại thời điểm t
  + = trọng lượng tại thời điểm t+1
  + = tốc độ học tại thời điểm t
  + = đạo hàm của hàm mất mát
  + = đạo hàm của trọng số tại thời điểm t
  + = tổng bình phương của các gradient trong quá khứ. [tức là tổng()] (ban đầu, = 0)
  + β = Tham số trung bình động (hằng số, 0.9)
  + = Hằng số dương nhỏ ()

Lợi ích khi sử dụng thuật toán tối ưu Adam:

* Hội tụ nhanh hơn :  Bằng cách điều chỉnh tốc độ học trong quá trình huấn luyện, Adam hội tụ nhanh hơn nhiều so với SGD.
* Dễ triển khai:  Chỉ yêu cầu độ dốc bậc nhất, Adam rất dễ triển khai và kết hợp với mạng lưới thần kinh sâu. Một vài dòng mã sử dụng Python và PyTorch là tất cả những gì ta cần.
* Thuật toán mạnh mẽ: Adam hoạt động tốt trên nhiều kiến ​​trúc mô hình khác nhau.
* Yêu cầu ít bộ nhớ:  Adam chỉ yêu cầu lưu trữ khoảnh khắc đầu tiên và thứ hai của độ dốc, giữ cho nhu cầu bộ nhớ ở mức thấp.
* Việc áp dụng rộng rãi trong cộng đồng: Adam được sử dụng rộng rãi bởi những người thực hành deep learning và đã trở thành một công cụ tối ưu hóa mặc định.

1. Triển khai

A diagram of a process

Description automatically generated

1. Thực nghiệm
2. Tập dữ liệu
3. Tiền xử lý

|  |
| --- |
|  |
|  |
|  |
|  |

1. Mô hình: Epoch – Accuracy

A graph with blue lines

Description automatically generated

1. Các bước triển khai trên phần cứng