|  |  |
| --- | --- |
| **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM** | |
| **Độc lập – Tự do – Hạnh phúc** |

**LUẬN VĂN THẠC SĨ**

**Thông tin đề tài**

1. Họ và tên học viên: Ngô Thị Thanh Bình SHHV: CB160533

2. Chuyên ngành: Công nghệ thông tin Lớp: CNTT 2016B

3. Người hướng dẫn:

* TS. Phạm Văn Hải, Viện Công nghệ thông tin và Truyền thông, trường Đại học Bách khoa Hà Nội

4. Đơn vị: Viện Công nghệ Thông tin - Truyền thông.

5. Tên đề tài (tiếng Việt): Ứng dụng Học máy trong Kiểm thử phần mềm

6. Tên đề tài (tiếng Anh): Apply Machine Learning in Software Testing

**CHƯƠNG 2: XÂY DỰNG HỆ THỐNG ỨNG DỤNG HỌC MÁY**

**VÀO KIỂM THỬ PHẦN MỀM**

1. **Mô tả bài toán**

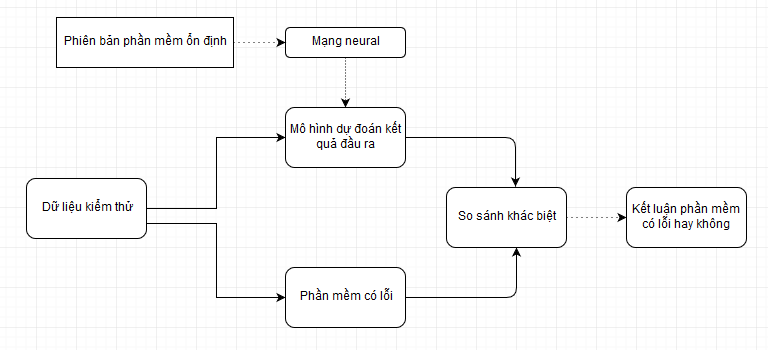
Khi thực hiện việc kiểm thử phần mềm, cơ bản nhất người ta cần xác định hai thứ: kết quả mong muốn và kết quả thực chạy trên phần mềm. Việc so sánh kết quả mong đợi với kết quả thực tế sẽ giúp phát hiện lỗi trên phần mềm.

Kết quả mong muốn này có được từ đặc tả yêu cầu phần mềm hoặc từ phiên bản chạy ổn định và đáng tin của phần mềm. Tuy nhiên điều này không phải lúc nào cũng dễ dàng. Trường hợp đầu tiên, đặc tả phần mềm thường sẽ chỉ bao gồm các yêu cầu mang tính ngữ nghĩa nhiều hơn là dữ liệu và kết quả cụ thể. Trường hợp thứ hai, phiên bản phần mềm đáng tin có thể không được giữ khi thay đổi môi trường, nâng cấp hệ thống,…

Do đó, bài toán ở đây là làm thế nào xây dựng một ứng dụng tự động tạo kết quả mong đợi cho phần mềm và các kết quả này cần có độ chính xác cao. Từ đó trợ giúp đánh giá lỗi phần mềm khi thực hiện hoạt động kiểm thử.

1. **Mô hình đề xuất**

Kiến trúc ứng dụng học máy xây dựng mô hình kiểm thử phẩn mềm được có thể được xây dựng như hình sau:



Hình 1 Mô hình kiến trúc tổng quát

**Đầu vào**: dữ liệu được chuẩn hóa để chạy trên cả mô hình dự đoán và phần mềm cần kiểm thử

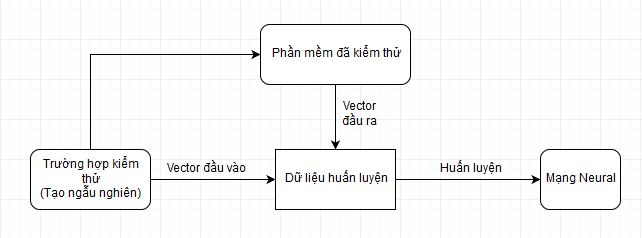
**Đầu ra**: độ khác biệt của kết quả trên phần mềm cần kiểm thử so với kết quả mô hình, trong đó:

* Kết quả khác biệt lớn: kết luận phần mềm có lỗi
* Kết quả tương đồng lớn: kết luận phần mềm không có lỗi
* Kết quả không tỏ rõ khác biệt hay tương đồng rõ rệt: kết luận mô hình không đáng tin và cần cải tiến.

Mô hình mạng neural đa lớp đã chứng tỏ khả năng giải quyết các bài toán thông dụng đến phức tạp một cách hiệu quả. Theo đó, phương pháp này được chọn để xây dựng mô hình dự đoán kết quả phần mềm cần kiểm thử.

1. **Diễn giải các bước**

* Xây dựng mạng neural
* Giai đoạn huấn luyện: Các vector đầu vào (trường hợp kiểm thử) được lựa chọn ngẫu nhiên, tương ứng với đặc tả yêu cầu của phần mềm cần kiểm thử. Mỗi đầu vào này được đưa vào phần mềm có kết quả chạy ổn định và đã được kiểm thử để có được đầu ra tương ứng. Bộ đầu vào – đầu ra này được sử dụng để huấn luyện mạng neural.



Hình 2 Tổng quát giai đoạn huấn luyện

* Giai đoạn kiểm tra: Vẫn là bộ dữ liệu tương tự như ở giai đoạn huấn luyện nhưng lấy từ phiên bản mới hơn của phần mềm để kiểm tra kết quả của mạng neural. Kiểm tra kết quả của mạng neural trên bộ dữ liệu này nếu cho sai số dự đoán thấp đạt mong đợi, có thể coi đã thành công tạo mô hình và đưa mô hình này vào sử dụng trong kiểm thử phần mềm.
* Module so sánh mạng neural và phần mềm có lỗi

Công cụ so sánh này được coi là một phương pháp độc lập để so sánh kết quả từ mạng neural và kết quả từ phiên bản phần mềm có lỗi. Việc so sánh này đương nhiên sẽ được thiết lập tự động để tránh kết quả bị ảnh hưởng bởi các yếu tố bên ngoài. Đồng thời, nó cũng thay thế những người kiểm thử thủ công có ý kiến chủ quan dựa trên kiến thức đã có về phần mềm.

Công cụ này về cơ bản sẽ đo khoảng cách tuyệt đối giữa kết quả của mô hình mạng neural và kết quả phần mềm có lỗi với cùng một đầu vào. Khoảng cách tuyệt đối này sẽ được giới hạn giá trị trong đoạn [0.0;1.0] và được dùng để xác định kết quả từ phần mềm là đúng hay lỗi.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Đầu ra phần mềm cần kiểm thử | |
|  |  | Đúng | Sai |
| Đầu ra mạng neural | Đúng | 1 True Positive | 2 True Negative |
| Sai | 4 False Negative | 3 False Positive |

Bảng 1 Phân loại các trường hợp đầu ra

Bảng 1 thể hiện bốn trường hợp tương ứng với các khả năng đầu ra của phần mềm và mạng neural. Có bốn khả năng như vậy là bởi mạng neural bản chất là mô phỏng của phần mềm thực sự nên chắc chắn không thể tránh khỏi một số ít kết quả sẽ không đúng. Về phía đầu ra của phần mềm cần kiểm thử đương nhiên sẽ có kết quả sai. Ví dụ nếu đầu ra của mạng neural đúng còn đầu ra của phần mềm kiểm thử sai ta có trường hợp True Negative, từ đó công cụ so sánh cho ra kết quả xa nhau và có thể kết luận phần mềm có lỗi thực sự. Tương tự như vậy, các trường hợp còn lại cũng sẽ dẫn tới các kết luận tương ứng. Hai trường hợp quan trọng cần quan tâm là 2 và 3 khi phần mềm kiểm thử cho kết quả sai. Ngoài ra cần lưu ý hai bộ trường hợp 1 và 3, 2 và 4 khi công cụ so sánh sẽ cho ra kết quả giống nhau.

1. Ví dụ ứng dụng

Thử nghiệm trên ứng dụng phê duyệt kết quả tín dụng

1. Tổng kết chương

**CHƯƠNG 3: CÀI ĐẶT VÀ THỰC NGHIỆM**

1. Cài đặt chương trình
2. Kết quả chương trình
3. Tổng kết chương

**CHƯƠNG 4: KẾT LUẬN**

1. Kết luận
2. Định hướng phát triển