|  |
| --- |
| **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**

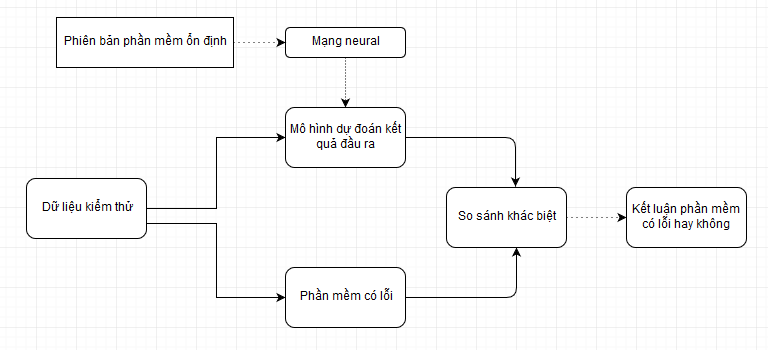
Khi thực hiện việc kiểm thử phần mềm, cơ bản nhất người ta cần xác định hai thứ: kết quả mong muốn và kết quả thực chạy trên phần mềm. Việc so sánh kết quả mong đợi với kết quả thực tế sẽ giúp phát hiện lỗi trên phần mềm.

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.

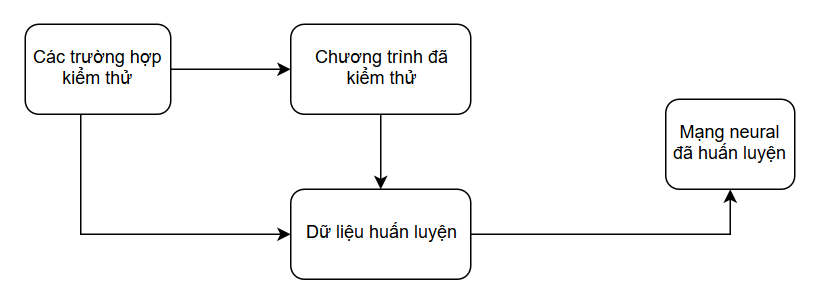
Mô hình mạng neural đa lớp đã chứng tỏ khả năng giải quyết các bài toán thông dụng đến phức tạp một cách hiệu quả. Theo đó, phương pháp này được chọn để xây dựng mô hình dự đoán kết quả phần mềm cần kiểm thử.

1. **Xây dựng mô hình**

Để có được mô hình kiểm thử phần mềm, đầu vào, đầu ra và định nghĩa nhãn phải được xác định

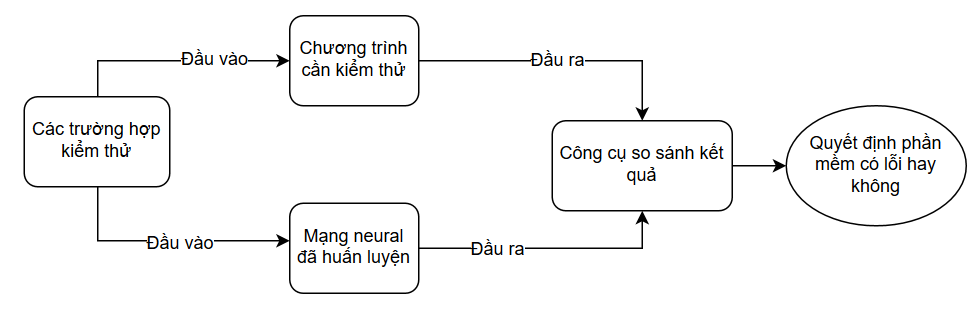
* 1. **Xây dựng dữ liệu**
  2. **Mạng neural**

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.



Hình 3 Tổng quát giai đoạn đánh giá

* 1. **Công cụ so sánh**
* 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. **Ứng dụng**

Thử nghiệm trên ứng dụng phê duyệt kết quả tín dụng

Thử nghiệm này sẽ trực tiếp triển khai mô hình đề xuất trên và được chia làm ba phần, đầu tiên là phần mềm mẫu, hai là mạng neural, và ba là công cụ so sánh và kết luận kết quả kiểm thử. Đặc tả yêu cầu cho phần mềm phê duyệt kết quả tín dụng được sử dụng để cung cấp đầu vào và đầu ra cũng như cài lỗi. Việc thiết kế mạng neural cũng phụ thuộc vào kiểu dữ liệu đầu vào đầu ra này. Hình 1 và hình 2 diễn tả quy trình thử nghiệm sẽ diễn ra. Trong giai đoạn huấn luyện, dữ liệu đầu vào dùng cho mạng neural và chương trình đã kiểm thử sẽ được tạo ra ngẫu nhiên. Trong giai đoạn đánh giá, một phần mềm có lỗi sẽ được tạo ra bằng cách thay đổi đoạn mã gốc trên phần mềm ban đầu.

*Thiết kế phần mềm phê duyệt kết quả tín dụng*

Phần mềm được sử dụng trong thử nghiệm là một phần mềm phê duyệt tín dụng đơn giản. Phần mềm này sẽ đọc các thông tin tổng hợp về khách hàng cần duyệt tín dụng và đưa ra hai kết luận. một là có duyệt tín dụng cho khách không, thứ hai là nếu duyệt thì khách hàng được cấp mức tín dụng bao nhiêu. Lập trình viên sẽ dựa theo các ràng buộc khác nhau đối với đầu vào để thiết kế thuật toán bám sát các ràng buộc đó. Phần mềm này có thể coi là có thể đại diện mức rộng và đủ, khi mà đầu ra chắc chắn sẽ phụ thuộc, có thể không phải tất cả, nhưng phần lớn ở đầu vào. Dữ liệu huấn luyện được tạo ra ngẫu nhiên sử dụng đặc tả phần mềm với mô tả chi tiết cho từng yếu tố ảnh hưởng tới đầu ra.

*Thiết kế mạng neural*

Mạng neural sử dụng trong thự nghiệm này sẽ dùng trực tiếp dữ liệu đầu vào của phần mềm phê duyệt tín dụng. Các đầu vào này đều được chuẩn hóa về dạng số. Có hai đầu ra thuộc hai loại khác nhau nên sẽ phải được xử lí khác nhau. Đầu ra cho việc có duyệt tín dụng hay không sẽ thuộc dạng hai lớp (0 và 1) còn đầu ra cho mức tín dụng là liên tục nhưng sẽ được xử lí để gom thành 10 nhãn (0 tới 9) và chuẩn hóa dựa trên mức cao nhất và thấp nhất.

*Thiết kế công cụ so sánh và kết luận*

Khi bắt đầu quá trình so sánh và kết luận, các lỗi sẽ lần lượt đưa vào phần mềm ban đầu. Các lỗi này có thể gây ảnh hưởng từ không đáng kể tới nghiêm trọng lên kết quả của phần mềm. Lúc này, công cụ so sánh sẽ mô phỏng lại quá trình kiểm thử phần mềm và kết luận phần mềm có lỗi hay không dựa trên kết quả chạy trên phần mềm đã bị sửa và kết quả từ mạng neural. Các trường hợp kiểm thử này đều là độc lập không liên quan hay phụ thuộc lẫn nhau.

1. Tổng kết chương

**CHƯƠNG 3: CÀI ĐẶT VÀ THỰC NGHIỆM**

1. Cài đặt chương trình
   1. Thiết kế ứng dụng thử nghiệm

Ứng dụng được thiết kế với thuật toán mô phỏng quá trình duyệt tín dụng dựa trên các ràng buộc với từng đặc điểm thu thập từ hồ sơ khách hàng. Chi tiết các đặc điểm này được mô tả trong Bảng 2.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Kiểu dữ liệu** | **Kiểu yếu tố** | **Diễn giải** |
| Mã khách hàng | Integer | Đầu vào |  |
| Quốc tịch | Integer | Đầu vào | 0: Mỹ  1: Khác |
| Bang | Integer | Đầu vào | 0: Florida  1: Khác |
| Tuổi | Integer | Đầu vào | 1-100 |
| Giới tính | Integer | Đầu vào | 0: Nữ  1: Nam |
| Vùng | Integer | Đầu vào | 0-6 cho các vùng ở Mỹ |
| Mức thu nhập | Integer | Đầu vào | 0: Thu nhập năm dưới 10k USD  1: Thu nhập năm >= 10k USD  2: Thu nhập năm >= 25k USD  3: Thu nhập năm >= 50k USD |
| Số người phụ thuộc | Integer | Đầu vào | 0-4 |
| Tình trạng hôn nhân | Integer | Đầu vào | 0: Độc thân  1: Đã kết hôn |
| Duyệt cho vay | Integer | Đầu ra | 0: Không  1: Có |
| Mức duyệt cho vay | Integer | Đầu ra | >= 0 |

Bảng 2 Đặc tả các yếu tố đầu vào đầu ra

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| citizenship | state | age | sex | region | income\_class | no\_depend | marital | approved | amount |
| 1 | 1 | 95 | 1 | 2 | 0 | 1 | 0 | 1 | 1300 |
| 1 | 1 | 82 | 1 | 5 | 3 | 1 | 0 | 0 | 0 |
| 1 | 1 | 21 | 0 | 0 | 0 | 4 | 0 | 1 | 1500 |
| 1 | 0 | 7 | 1 | 6 | 2 | 2 | 0 | 0 | 0 |
| 1 | 0 | 81 | 0 | 2 | 2 | 1 | 0 | 1 | 2800 |
| 0 | 0 | 7 | 0 | 3 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 7 | 0 | 2 | 2 | 0 | 0 | 0 | 0 |
| 0 | 0 | 11 | 0 | 3 | 3 | 0 | 0 | 0 | 0 |
| 1 | 1 | 25 | 0 | 0 | 1 | 0 | 1 | 1 | 2200 |

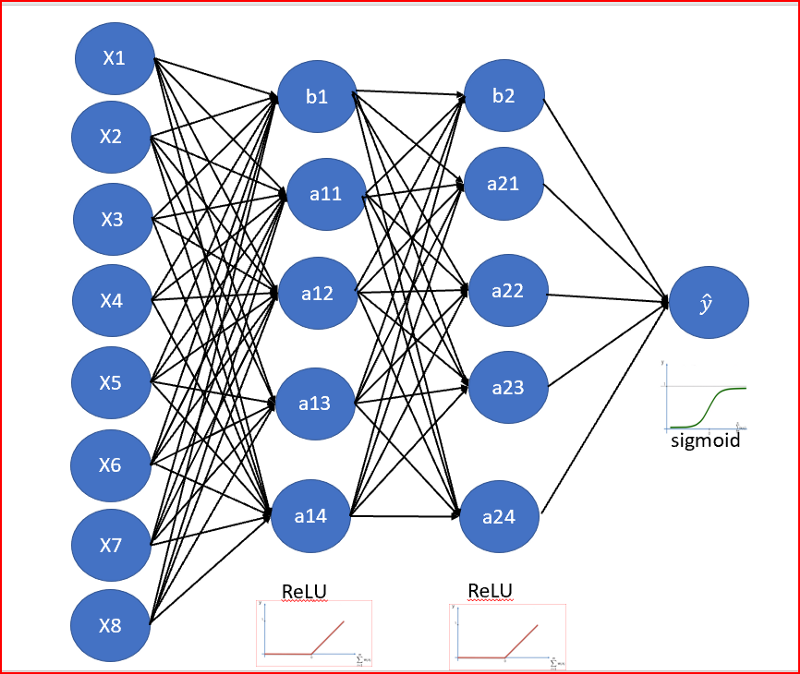
Bảng 3 Dữ liệu ví dụ

Lấy ví dụ, khách hàng đầu tiên trong bảng, là người Mỹ sống ở bang Florida, Nam giới 95 tuổi đến từ vùng 2 ở Mỹ, thuộc mức thu nhập 0 (dưới 10 000 USD/năm) với 1 người phụ thuộc tình trạng hôn nhân là độc thân. Khách này được duyệt tín dụng với hạn mức là 1300 USD. Vì mạng neural chỉ có thể nhận đầu vào dưới dạng số, các đầu vào có tính phân loại (quốc tịch, bang,…) được chuyển sang dạng số.

* 1. Huấn luyện mạng neural mô phỏng phần mềm

Mô hình mạng neural được phát triển trong luận văn sử dụng thư viện keras được xây dựng trên nền tensorflow của Google. Thư viện được sử dụng trên môi trường ảo dựng trên máy tính. Thư viện này cùng với môi trường đã có sẵn các

Tất cả dữ liệu (huấn luyện và kiểm thử) bao gồm 10000 bản ghi (trường hợp kiểm thử) được chia với tỉ lệ 80/20. Một tập dữ liệu khác được dùng để phát hiện lỗi trong phần mềm cần kiểm thử.



Thiết kế mạng neural trong luận văn này có kiến trúc kết nối đầy đủ với 3 lớp (hai lớp ẩn với lần lượt 100 và 50 nút, một lớp đầu ra, không tính đầu vào). Con số lớp và số nút trong từng lớp được lựa chọn ngẫu nhiên và thay đổi và tối ưu trong quá trình huấn luyện mạng. Lý tưởng mà nói cần một mạng đủ rộng để hiểu được xu hướng/ cấu trúc của dữ liệu. Activation function dùng cho hai lớp ẩn là ReLU còn lớp đầu ra là sigmoid.

Sau khoảng 200 epochs mô hình đạt được tỉ lệ chính xác khoảng 99% (hai lớp) và \_\_\_ (đa lớp)

* 1. Công cụ so sánh và xác định lỗi phần mềm

1. Kết quả chương trình
2. Tổng kết chương

**CHƯƠNG 4: KẾT LUẬN**

1. Kết luận
2. Định hướng phát triển