**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**DEEP LEARNING**

*Người thực hiện*: **NGUYỄN QUANG BẢO – MSHV:20083601**

**NGUYỄN XUÂN GIANG - MSSV: 20079601**

Lớp **: 420300411601**

Khoá  **: 16**

*Người hướng dẫn*: **TS BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**DEEP LEARNING**

Người thực hiện: **NGUYỄN QUANG BẢO – MSHV:20083601**

**NGUYỄN XUÂN GIANG - MSSV: 20079601**

Lớp **: 420300411601**

Khoá  **: 16**

Người hướng dẫn: **TS. BÙI THANH HÙNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Chúng tôi xin gửi lời cảm ơn chân thành nhất đế quý thầy TS.Bùi Thanh Hùng – Giảng viên đã tận tình hướng dẫn cho chúng tôi trong suốt quá trình học tập môn Xử lý ngôn ngữ tự nhiên và tạo điều kiện cho chúng tôi làm Đồ án này. Dưới sự hướng dẫn của Thầy chúng tôi được tiếp cận với những kiến thức chuyên môn sâu, rộng trong lĩnh vực xử lý ngôn ngữ tự nhiên. Qua đó có thể hoàn thành Đồ án này một cách thuận lợi.

Một lần nữa chúng tôi xin gửi lời cảm ơn sâu sắc đến Thầy. Chúc Thầy thật nhiều sức khỏe, thành công trong công việc và trong cuộc sống. Xin chân thành cảm ơn Thầy.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS. Bùi Thanh Hùng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Công nghiệp TP Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Quang Bảo*

*Nguyễn Xuân Giang*

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Bài toán Dự đoán chuỗi thời gian (Time Series Prediction) và Phân loại chuỗi thời gian (Time Series Classification).Bài toán Dự đoán chuỗi thời gian liên quan đến việc sử dụng dữ liệu theo thời gian để dự đoán giá trị tương lai của chuỗi đó, bài toán Phân loại chuỗi thời gian đặt ra thách thức trong việc gán một nhãn (label) cho mỗi phần tử trong chuỗi thời gian.Trong những nghiên cứu về bài toán này trước đây Trong những nghiên cứu về bài toán này trước đây để đa số đều sử dụng đến các phương pháp heuristics với một hàm đánh giá từ đó tìm ra chỉ số tương đồng (similarity) giữa các thực thể từ đó dự đoán khả năng tồn tại liên kết giữa chúng. Tuy nhiên các phương pháp heuristics đều dựa trên những giả định về sự tồn tại liên kết giữa các thực thể, điều này dẫn đến khi những giả định này không chính xác thì kết quả của thuật toán bị giảm đi đáng kể. Thay vì sử dụng những phương pháp trên, phương pháp sử dụng mạng neural nhân tạo tỏ ra hứa hẹn hơn trong việc giải những bài toán về mạng này. Đã có một số công trình áp dụng cấu trúc đồ thị vào mạng neural nhân tạo như là mạng neural LSTM (Long Short-Term Memory) và GRU (Gated Recurrent Unit) là hai loại mạng nơ-ron nhân tạo, đặc biệt được thiết kế để xử lý chuỗi dữ liệu và vấn đề vanishing gradient trong mô hình RNN (Recurrent Neural Network) . Các kết quả thử nghiệm đã cho thấy, phương pháp sử dụng mạng neural đồ thị này là đặc biệt khả quan.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc153149646)

[PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc153149647)

[TÓM TẮT iv](#_Toc153149648)

[MỤC LỤC 1](#_Toc153149649)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 2](#_Toc153149650)

[DANH MỤC CÁC HÌNH VẼ 3](#_Toc153149651)

[DANH MỤC CÁC BẢNG 4](#_Toc153149652)

[1.1 Giới thiệu về bài toán 5](#_Toc153149653)

[1.2 Phân tích yêu cầu của bài toán 6](#_Toc153149654)

[1.2.1 Yêu cầu của bài toán 6](#_Toc153149655)

[1.2.2 Các phương pháp giải quyết bài toán 6](#_Toc153149656)

[1.2.3 Phương pháp đề xuất giải quyết bài toán 7](#_Toc153149657)

[1.3 Phương pháp giải quyết bài toán 16](#_Toc153149658)

[1.3.1 Mô hình tổng quát 16](#_Toc153149659)

[1.3.2 Đặc trưng của mô hình đề xuất 16](#_Toc153149660)

[1.4 Thực nghiệm 19](#_Toc153149661)

[1.4.1 Dữ liệu 19](#_Toc153149662)

[1.4.2 Xử lý dữ liệu 20](#_Toc153149663)

[1.4.3 Công nghệ sử dụng 22](#_Toc153149664)

[1.4.4 Cách đánh giá 23](#_Toc153149665)

[1.5 Kết quả đạt được 24](#_Toc153149666)

[1.6 Kết luận 27](#_Toc153149667)

[LÀM VIỆC NHÓM 28](#_Toc153149668)

[TÀI LIỆU THAM KHẢO 29](#_Toc153149669)

[TỰ ĐÁNH GIÁ 31](#_Toc153149670)

# DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

GRU Gated Recurrent Unit

RNN Recurrent Neural Network

CNN Convolutional Neural Network

LSTM Long Short-Term Memory

DANH MỤC CÁC HÌNH VẼ

**[Hình 1:](#_Toc153144420)** [Giới thiệu kiến trúc GRU 10](#_Toc153144420)

**[Hình 2:](#_Toc153144421)** [Giới thiệu kiến trúc LSTM 12](#_Toc153144421)

**[Hình 3:](#_Toc153144422)** [Giới thiệu kiến trúc SimpleRNN 15](#_Toc153144422)

**[Hình 4:](file:///C:\\Users\\hungp\\OneDrive%20-%20Industrial%20University%20of%20HoChiMinh%20City\\Desktop\\deepln.docx" \l "_Toc153144423)** [Mô hình tổng quát 17](file:///C:\\Users\\hungp\\OneDrive%20-%20Industrial%20University%20of%20HoChiMinh%20City\\Desktop\\deepln.docx" \l "_Toc153144423)

**[Hình 5:](#_Toc153144424)** [Accuracy giữa các mô hình 25](#_Toc153144424)

**[Hình 6:](#_Toc153144425)** [Loss giữa các mô hình 26](#_Toc153144425)

**[Hình 7:](#_Toc153144426)** [Average Precision Score giữa các mô hình 26](#_Toc153144426)

DANH MỤC CÁC BẢNG

**[Bảng 1:](#_Toc153144591)** [Công nghệ sử dụng 22](#_Toc153144591)

**[Bảng 2:](#_Toc153144592)** [Các mẫu dự đoán 26](#_Toc153144592)

**[Bảng 3:](#_Toc153144593)** [Kết quả thực nghiệm 26](#_Toc153144593)

**[Bảng 4:](#_Toc153144594)** [Kết quả so sánh 26](#_Toc153144594)

* 1. Giới thiệu về bài toán

Mục tiêu của cuộc thi này là phát hiện thời điểm bắt đầu và thức giấc trong giấc ngủ. Bạn sẽ phát triển một mô hình được đào tạo về dữ liệu gia tốc kế đeo ở cổ tay để xác định trạng thái ngủ của một người.

Tập dữ liệu bao gồm khoảng 500 bản ghi dữ liệu gia tốc kế đeo ở cổ tay trong nhiều ngày được chú thích bằng hai loại sự kiện: khởi phát , bắt đầu giấc ngủ và thức dậy , kết thúc giấc ngủ.

Mặc dù nhật ký giấc ngủ vẫn là tiêu chuẩn vàng nhưng khi làm việc với dữ liệu gia tốc kế, chúng tôi coi giấc ngủ là khoảng thời gian không hoạt động dài nhất khi đang đeo đồng hồ. Đối với dữ liệu này, chúng tôi đã hướng dẫn người xếp hạng một số hướng dẫn cụ thể:

- Một lần ngủ phải kéo dài ít nhất 30 phút

- Một khoảng thời gian ngủ có thể bị gián đoạn bởi các hoạt động không quá 30 phút liên tục

- Không thể phát hiện cửa sổ ngủ trừ khi đồng hồ được cho là đã được đeo trong thời gian đó (được trình bày chi tiết bên dưới)

- Khoảng thời gian ngủ dài nhất trong đêm là khoảng thời gian duy nhất được ghi lại

- Nếu không có khoảng thời gian ngủ hợp lệ nào có thể nhận dạng được thì cả sự kiện bắt đầu và sự kiện đánh thức đều không được ghi lại cho đêm đó.

Các sự kiện về giấc ngủ không cần phải xếp chồng lên nhau trong ngày và do đó không có quy tắc cứng nhắc nào xác định số lượng có thể xảy ra trong một khoảng thời gian nhất định. Tuy nhiên, không nên chỉ định nhiều hơn một cửa sổ mỗi đêm. Ví dụ: một cá nhân có thể có thời gian ngủ từ 01h00–06h00 và 19h00–23h30 trong cùng một ngày dương lịch, mặc dù được chỉ định cho các đêm liên tiếp

Số đêm được ghi lại cho một loạt phim gần bằng số khoảng thời gian 24 giờ trong loạt phim đó.

Mặc dù mỗi chuỗi là một bản ghi liên tục nhưng có thể có những khoảng thời gian trong chuỗi khi thiết bị đo gia tốc bị tháo ra. Những khoảng thời gian này được xác định là những khoảng thời gian mà tín hiệu gia tốc kế có sự thay đổi nhỏ đáng ngờ trong một khoảng thời gian dài, điều này không thực tế đối với những người tham gia thông thường.

Công việc của bạn có thể giúp các nhà nghiên cứu có thể tiến hành các nghiên cứu về giấc ngủ quy mô lớn hơn, đáng tin cậy hơn trên nhiều nhóm dân cư và bối cảnh khác nhau. Kết quả của những nghiên cứu như vậy có thể cung cấp nhiều thông tin hơn về giấc ngủ.

Kết quả thành công của cuộc thi này cũng có thể có ý nghĩa quan trọng đối với trẻ em và thanh thiếu niên, đặc biệt là những trẻ có khó khăn về tâm trạng và hành vi.

1.2 Phân tích yêu cầu của bài toán

1.2.1 Yêu cầu của bài toán

Yêu cầu của bài toán này là phát hiện thời điểm bắt đầu và thức giấc trong giấc ngủ. Bạn sẽ phát triển một mô hình được đào tạo về dữ liệu gia tốc kế đeo ở cổ tay để xác định trạng thái ngủ của một người.

1.2.2 Các phương pháp giải quyết bài toán

Bài toán phân loại giấc ngủ sử dụng dữ liệu gia tốc kế đeo trên cổ tay là một bài toán quan trọng và thách thức trong nghiên cứu về giấc ngủ và sức khỏe. Có nhiều phương pháp đã được đề xuất để giải quyết bài toán này, tuy nhiên không có phương pháp nào là tối ưu và hoàn hảo. Dưới đây là một số phương pháp tiêu biểu:

Phương pháp học máy

Phương pháp sử dụng Random forests1: Phương pháp này được đề xuất bởi Sundararajan et al. (2020)[1] trong bài báo Sleep classification from wrist-worn accelerometer data using random forests.

Phương pháp học sâu

Phương pháp sử dụng mạng nơ-ron hồi : Phương pháp này được đề xuất bởi Chen et al. (2021) trong bài báo Three-level Sleep Stage Classification Based on Wrist-worn Accelerometer Data Using Recurrent Neural Networks[2]. Phương pháp này sử dụng các mạng nơ-ron hồi quy (RNN) để phân loại giấc ngủ thành ba cấp độ: ngủ sâu, ngủ nhẹ và thức.

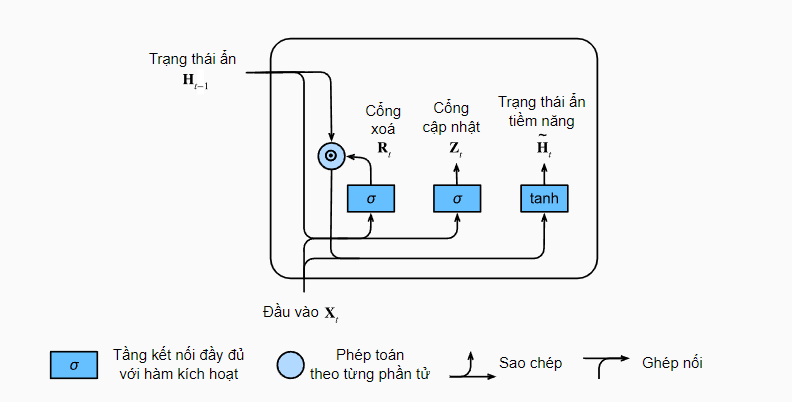
Phương pháp sử dụng mạng nơ-ron tích chập: Phương pháp này được đề xuất bởi Kortelainen et al. (2020) trong bài báo Estimation of respiration rate and sleeping position using a wearable accelerometer[3]. Phương pháp này sử dụng một mạng nơ-ron tích chập (CNN) để ước tính tần số hô hấp và tư thế ngủ sử dụng dữ liệu gia tốc kế đeo trên ngực.

1.2.3 Phương pháp đề xuất giải quyết bài toán

Như đã trình bày ở phần trên, trong bài toán này, đề xuất sử dụng các phương pháp học sâu để giải quyết. Các phương pháp học sâu sẽ phù hợp với các mô hình có dữ liệu đầu vào lớn và dữ liệu gia tốc kế có tính chất thời gian và không gian có thể được biểu diễn dưới dạng các chuỗi và các ma trận. Recurrent Neural Network (RNN)[4] là một loại mạng nơ-ron nhân tạo (neural network) được sử dụng rộng rãi trong các bài toán xử lý dữ liệu dạng chuỗi, như văn bản, âm thanh, video, v.v. RNN có thể học được các mối quan hệ thời gian giữa các phần tử trong một chuỗi, và sử dụng các đặc trưng này để phân loại, dự báo, hay thực hiện các tác vụ khác trên dữ liệu phù hợp cho bài toán phân loại giấc ngủ sử dụng dữ liệu gia tốc kế đeo trên cổ tay trong đồ án này. RNN có một số điểm nổi bật sau: Khả năng lưu lại thông tin từ các bước tính toán trước: RNN có thể sử dụng một bộ nhớ để lưu lại thông tin từ các bước tính toán trước đó, và sử dụng nó để ảnh hưởng đến các bước tính toán hiện tại. Điều này giúp RNN có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu. Khả năng xử lý các dữ liệu có độ dài khác nhau: RNN có thể xử lý các dữ liệu có độ dài khác nhau, bằng cách sử dụng các cấu trúc mạng khác nhau, chẳng hạn như many-to-one, one-to-many, many-to-many, v.v. Điều này giúp RNN có thể thích ứng với nhiều loại bài toán khác nhau. Khả năng kết hợp với các mạng nơ-ron khác: RNN có thể kết hợp với các mạng nơ-ron khác, chẳng hạn như CNN[5], để tận dụng được cả hai khía cạnh của dữ liệu, là thời gian và không gian. Điều này giúp RNN có thể xử lý các dữ liệu phức tạp và đa dạng hơn. Trong đồ án này, sử dụng các biến thể của RNN và mô hình GRU để giải quyết bài toán và tiến hành so sánh kết quả. Các biến thể của RNN và mô hình GRU được sử dụng trong bài toán bao gồm:

GRU là viết tắt của Gated Recurrent Unit, là một loại mạng nơ-ron hồi quy nâng cao, được phát triển bởi Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk và Yoshua Bengio vào năm 2014. Mô hình này được công bố lần đầu tiên trong bài báo “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation” tại Hội nghị Quốc tế về Máy học Thống kê (EMNLP) năm 2014.

Kiến trúc của GRU: GRU có kiến trúc gồm một lớp ẩn (hidden layer) và hai cổng, là cổng cập nhật (update gate) và cổng xóa (reset gate). Cổng cập nhật giúp quyết định thông tin nào từ lớp ẩn trước đó sẽ được giữ lại và thông tin nào sẽ được cập nhật. Cổng xóa giúp quyết định thông tin nào từ lớp ẩn trước đó sẽ được sử dụng để tính toán trạng thái ẩn hiện tại.



**Hình 1:** Giới thiệu kiến trúc GRU

Hiệu năng của GRU: GRU có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, và có thể tránh được hiện tượng biến mất đạo hàm (vanishing gradient) khi huấn luyện. GRU có độ phức tạp thấp hơn LSTM, nhưng vẫn có hiệu năng tương đương hoặc cao hơn trong nhiều bài toán.

Ứng dụng của GRU: GRU có nhiều ứng dụng trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dự báo chuỗi thời gian, v.v. Một số ví dụ về các bài toán sử dụng GRU là:

* Phân loại cảm xúc: GRU có thể được sử dụng để phân loại cảm xúc của người nói hoặc người viết, dựa trên nội dung và ngữ điệu của họ.
* Dịch máy: GRU có thể được sử dụng để dịch các văn bản từ một ngôn ngữ sang một ngôn ngữ khác, bằng cách sử dụng một mô hình mã hóa-giải mã (encoder-decoder).
* Tổng hợp giọng nói: GRU có thể được sử dụng để tạo ra các giọng nói tự nhiên từ các văn bản, bằng cách sử dụng một mô hình dựa trên sóng (wave-based model).

Ưu điểm của GRU:

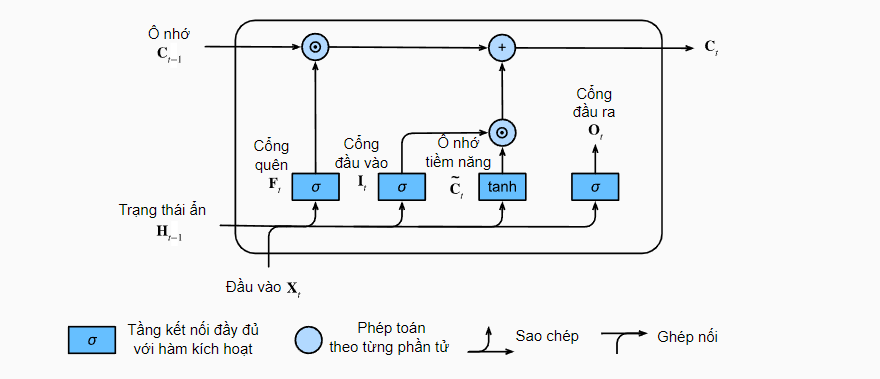
* Một số ưu điểm của GRU là:Khả năng học các mối quan hệ dài hạn và ngữ cảnh của dữ liệu: GRU có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, bằng cách sử dụng các cổng để điều chỉnh việc lưu lại và quên đi thông tin từ các bước tính toán trước.
* Khả năng tránh hiện tượng biến mất đạo hàm: GRU có thể tránh được hiện tượng biến mất đạo hàm khi huấn luyện, bằng cách sử dụng các cổng để kiểm soát luồng thông tin qua chuỗi trình tự.
* Độ phức tạp thấp hơn LSTM: GRU có độ phức tạp thấp hơn LSTM, vì nó chỉ có hai cổng, trong khi LSTM có ba cổng. Điều này giúp GRU tiết kiệm thời gian và tài nguyên khi huấn luyện và sử dụng.

Nhược điểm của GRU:

* Một số nhược điểm của GRU là:Khả năng học các mối quan hệ rất dài hạn kém hơn LSTM: GRU có thể học được các mối quan hệ dài hạn, nhưng không tốt bằng LSTM, vì nó không có cổng quên (forget gate) như LSTM. Cổng quên giúp LSTM có thể quyết định thông tin nào sẽ được quên đi hoàn toàn, và thông tin nào sẽ được giữ lại lâu dài.
* Khả năng thích ứng với các bài toán khác nhau kém hơn LSTM: GRU có thể thích ứng với các bài toán khác nhau, nhưng không tốt bằng LSTM, vì nó không có cổng đầu ra (output gate) như LSTM. Cổng đầu ra giúp LSTM có thể quyết định thông tin nào sẽ được đưa ra làm đầu ra, và thông tin nào sẽ được giữ lại trong bộ nhớ.

Kết luận: GRU là một loại mạng nơ-ron hồi quy nâng cao, có thể xử lý các dữ liệu dạng chuỗi, nhưng có những cải tiến so với RNN cơ bản. GRU có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, và có thể tránh được hiện tượng biến mất đạo hàm khi huấn luyện. GRU có độ phức tạp thấp hơn LSTM, nhưng vẫn có hiệu năng tương đương hoặc cao hơn trong nhiều bài toán. GRU có nhiều ứng dụng trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dự báo chuỗi thời gian, v.v.

LSTM [6] là viết tắt của Long Short-Term Memory, là một loại mạng nơ-ron hồi quy nâng cao, có thể xử lý các dữ liệu dạng chuỗi, như văn bản, âm thanh, video, v.v. LSTM được thiết kế để khắc phục vấn đề phụ thuộc dài hạn mà các mạng nơ-ron hồi quy cơ bản (RNN) gặp phải. LSTM có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, bằng cách sử dụng một bộ nhớ đặc biệt được gọi là trạng thái tế bào (cell state). LSTM cũng có ba loại cổng, là cổng quên (forget gate), cổng đầu vào (input gate) và cổng đầu ra (output gate), giúp điều chỉnh việc lưu trữ, cập nhật và truy xuất thông tin từ trạng thái tế bào.



**Hình 2:** Giới thiệu kiến trúc LSTM

LSTM có nhiều ứng dụng trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dự báo chuỗi thời gian, v.v. Một số ví dụ về các bài toán sử dụng LSTM là:

* Phân loại cảm xúc: LSTM có thể được sử dụng để phân loại cảm xúc của người nói hoặc người viết, dựa trên nội dung và ngữ điệu của họ.
* Dịch máy: LSTM có thể được sử dụng để dịch các văn bản từ một ngôn ngữ sang một ngôn ngữ khác, bằng cách sử dụng một mô hình mã hóa-giải mã (encoder-decoder).
* Tổng hợp giọng nói: LSTM có thể được sử dụng để tạo ra các giọng nói tự nhiên từ các văn bản, bằng cách sử dụng một mô hình dựa trên sóng (wave-based model)

Một số ưu điểm của LSTM là:

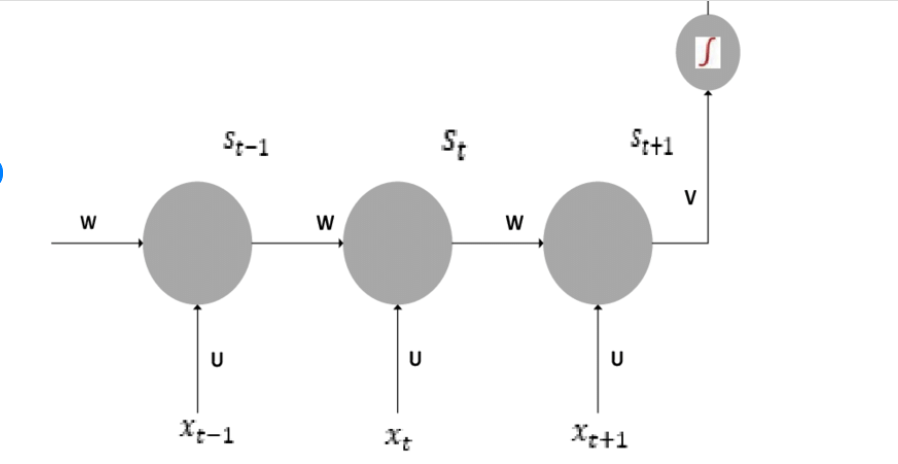
* Khả năng học các mối quan hệ dài hạn và ngữ cảnh của dữ liệu: LSTM có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, bằng cách sử dụng trạng thái tế bào để lưu trữ thông tin quan trọng và bỏ qua thông tin không cần thiết.
* Khả năng tránh hiện tượng biến mất đạo hàm: LSTM có thể tránh được hiện tượng biến mất đạo hàm khi huấn luyện, bằng cách sử dụng các cổng để kiểm soát luồng thông tin qua chuỗi trình tự.
* Khả năng thích ứng với các bài toán khác nhau: LSTM có thể thích ứng với các bài toán khác nhau, bằng cách sử dụng các cấu trúc mạng khác nhau, chẳng hạn như many-to-one, one-to-many, many-to-many, v.v.

Một số nhược điểm của LSTM là:

* Kiến trúc phức tạp: LSTM có kiến trúc phức tạp, đòi hỏi nhiều dữ liệu và thời gian để huấn luyện.
* Kích thước lớn: LSTM có kích thước lớn, đòi hỏi nhiều tài nguyên để lưu trữ và sử dụng.
* Khả năng học các mối quan hệ rất dài hạn kém hơn các mô hình khác: LSTM có thể học được các mối quan hệ dài hạn, nhưng không tốt bằng các mô hình khác, chẳng hạn như Transformer, có thể sử dụng cơ chế chú ý (attention) để tăng cường khả năng học các mối quan hệ rất dài hạn.

Kết luận: LSTM là một loại mạng nơ-ron hồi quy nâng cao, có thể xử lý các dữ liệu dạng chuỗi, nhưng có những cải tiến so với RNN cơ bản. LSTM có thể học được các mối quan hệ dài hạn và ngữ cảnh của dữ liệu, bằng cách sử dụng một bộ nhớ đặc biệt và ba loại cổng. LSTM có nhiều ứng dụng trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dự báo chuỗi thời gian, v.v. Tuy nhiên, LSTM cũng có một số nhược điểm, như kiến trúc phức tạp, kích thước lớn và khả năng học các mối quan hệ rất dài hạn kém hơn các mô hình khác.

SimpleRNN[7] là một loại mạng nơ-ron hồi quy cơ bản, có thể xử lý các dữ liệu dạng chuỗi, như văn bản, âm thanh, video, v.v. SimpleRNN có kiến trúc gồm một lớp ẩn (hidden layer) và một lớp đầu ra (output layer). Mỗi lớp ẩn nhận đầu vào là dữ liệu tại bước thời gian hiện tại (x\_t) và trạng thái ẩn của bước thời gian trước đó (h\_ {t-1}). Lớp ẩn sử dụng một hàm kích hoạt (thường là tanh hoặc ReLU) để tính toán trạng thái ẩn hiện tại (h\_t). Lớp đầu ra sử dụng một hàm kích hoạt khác (thường là softmax) để tính toán đầu ra tại bước thời gian hiện tại (y\_t).



Hình 3: Giới thiệu kiến trúc SimpleRNN

SimpleRNN có nhiều ứng dụng trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dự báo chuỗi thời gian, v.v. Một số ví dụ về các bài toán sử dụng Simple RNN là:

* Phân loại văn bản: SimpleRNN có thể được sử dụng để phân loại văn bản theo các nhãn như thể loại, tác giả, cảm xúc, v.v.
* Sinh văn bản: SimpleRNN có thể được sử dụng để sinh ra các văn bản mới, dựa trên một văn bản gốc hoặc một từ khóa.
* Dự báo giá cổ phiếu: Simple RNN có thể được sử dụng để dự báo giá cổ phiếu trong tương lai, dựa trên các dữ liệu lịch sử.

Một số ưu điểm của Simple RNN là:

* Khả năng xử lý các dữ liệu dạng chuỗi: Simple RNN có thể xử lý các dữ liệu dạng chuỗi, bằng cách sử dụng trạng thái ẩn để lưu trữ thông tin từ các bước thời gian trước.
* Khả năng học các mối quan hệ ngắn hạn của dữ liệu: Simple RNN có thể học được các mối quan hệ ngắn hạn của dữ liệu, bằng cách sử dụng hàm kích hoạt để tính toán trạng thái ẩn và đầu ra.
* Độ phức tạp thấp: Simple RNN có độ phức tạp thấp, vì nó chỉ có một lớp ẩn và một lớp đầu ra, và chỉ sử dụng một ma trận trọng số cho mỗi lớp.

Một số nhược điểm của Simple RNN là:

* Khả năng học các mối quan hệ dài hạn của dữ liệu kém: Simple RNN có khả năng học các mối quan hệ dài hạn của dữ liệu kém, vì nó bị ảnh hưởng bởi hiện tượng biến mất đạo hàm (vanishing gradient) khi huấn luyện. Hiện tượng này xảy ra khi đạo hàm của hàm kích hoạt bị giảm dần theo thời gian, làm cho các trọng số của các bước thời gian xa không được cập nhật hiệu quả.
* Khả năng thích ứng với các bài toán khác nhau kém: Simple RNN có khả năng thích ứng với các bài toán khác nhau kém, vì nó không có cơ chế điều chỉnh việc lưu trữ, cập nhật và truy xuất thông tin từ trạng thái ẩn. Các loại mạng nơ-ron hồi quy nâng cao, như LSTM hoặc GRU, có thể sử dụng các cổng để kiểm soát luồng thông tin qua chuỗi trình tự.

Kết luận: SimpleRNN là một loại mạng nơ-ron hồi quy cơ bản, có thể xử lý các dữ liệu dạng chuỗi, nhưng có những hạn chế so với các loại mạng nơ-ron hồi quy nâng cao. Simple RNN có thể học được các mối quan hệ ngắn hạn của dữ liệu, bằng cách sử dụng một lớp ẩn và một lớp đầu ra. SimpleRNN có nhiều ứng dụng trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói, dự báo chuỗi thời gian, v.v. Tuy nhiên, Simple RNN cũng có một số nhược điểm, như khả năng học các mối quan hệ dài hạn của dữ liệu kém, khả năng thích ứng với các bài toán khác nhau kém.

1.3 Phương pháp giải quyết bài toán

1.3.1 Mô hình tổng quát

Hình 4: Mô hình tổng quát

Deep Learning Model

-----------------------------------

GRU

LSTM

SIMPLERNN

Data Pre-Processing

Separate Training and Test Data

Online Dataset

Test Data

Training Data

Evalute Model

Build Model

Output

Wakeup

Onset

1.3.2 Đặc trưng của mô hình đề xuất

Deep Learning Model

Sau khi có dữ liệu thích hợp thì thực hiện việc huấn luyện mô hình đã đề xuất ở trên là GRU ,SimpleRNN,LSTM.

Mô hình GRU là một loại mạng nơ-ron hồi quy (RNN) có thể học các phụ thuộc dài hạn trong chuỗi. Điều này khiến nó trở nên phù hợp cho các bài toán phân loại và dự đoán liên quan đến chuỗi dữ liệu.Trong bài toán phân loại văn bản, mô hình GRU có thể được sử dụng để học các mối quan hệ giữa các từ trong một văn bản. Điều này cho phép mô hình GRU xác định các đặc điểm của văn bản có liên quan đến phân loại của nó.Trong bài toán dự đoán chuỗi thời gian, mô hình GRU có thể được sử dụng để học các xu hướng và mô hình trong dữ liệu chuỗi thời gian. Điều này cho phép mô hình GRU dự đoán các giá trị trong tương lai của chuỗi thời gian.

Lớp đầu vào là một tensor 2D có hình dạng (TIMESTEPS, NBR\_FEATURES), trong đó TIMESTEPS là số bước thời gian trong chuỗi đầu vào và NBR\_FEATURES là số tính năng trong mỗi bước thời gian.

Lớp tích chập là một lớp tích chập 1D với 32 bộ lọc, kích thước hạt là 8 và bước là 1. Hàm kích hoạt cho lớp tích chập là 'relu'. Chèn cho lớp tích chập là 'same', có nghĩa là chuỗi đầu ra sẽ có cùng độ dài với chuỗi đầu vào.

Lớp GRU hai chiều là một lớp GRU hai chiều với TIMESTEPS đơn vị hồi quy. Hàm kích hoạt cho lớp GRU là 'tanh'. Tham số return\_sequences được đặt thành True, có nghĩa là đầu ra của lớp GRU là một chuỗi vectơ, thay vì một vectơ duy nhất.

Lớp đầu ra là một lớp dày 1D với 1 đơn vị đầu ra. Hàm kích hoạt cho lớp đầu ra là 'sigmoid'.

Nhìn chung, mô hình GRU là một công cụ mạnh mẽ có thể được sử dụng cho nhiều tác vụ phân loại và dự đoán. Mô hình này có thể học các phụ thuộc dài hạn trong chuỗi, điều này khiến nó trở nên phù hợp cho các tác vụ liên quan đến chuỗi dữ liệu.

Mô hình SimpleRNN, hay mạng nơ-ron hồi quy đơn giản, là một loại mạng nơ-ron hồi quy (RNN) được sử dụng để xử lý dữ liệu tuần tự. Nó là một phiên bản đơn giản hóa của mạng LSTM (Lưu trữ ngắn hạn dài) được sử dụng phổ biến hơn, nhưng nó vẫn có thể học các phụ thuộc dài hạn trong chuỗi.Kiến trúc mô hình:

Lớp tích chập đóng vai trò là bộ trích xuất tính năng, trích xuất thông tin liên quan từ chuỗi đầu vào.Nó sử dụng một lớp tích chập 1D với 32 bộ lọc, kích thước hạt là 8 và bước là 1.Hàm kích hoạt 'relu' đưa ra tính phi tuyến tính, cho phép mô hình học các mẫu phức tạp.Chèn 'same' đảm bảo rằng chuỗi đầu ra có cùng độ dài với chuỗi đầu vào.

Hai lớp SimpleRNN được sử dụng để nắm bắt các phụ thuộc dài hạn trong chuỗi.Mỗi lớp SimpleRNN bao gồm TIMESTEPS[8] đơn vị hồi quy.

Hàm kích hoạt 'tanh' được áp dụng cho đầu ra của mỗi lớp SimpleRNN.Tham số return\_sequences được đặt thành True, chỉ ra rằng đầu ra của mỗi lớp SimpleRNN là một chuỗi vectơ.

Lớp đầu ra bao gồm một đơn vị dày 1D với hàm kích hoạt 'sigmoid'.Đầu ra đại diện cho xác suất của một sự kiện cụ thể xảy ra, phù hợp cho các tác vụ phân loại nhị phân.

SimpleRNN là một công cụ mạnh mẽ cho các bài toán phân loại và dự đoán liên quan đến dữ liệu tuần tự. Tính đơn giản, tính toán hiệu quả và khả năng học các phụ thuộc dài hạn của nó khiến nó trở thành lựa chọn có giá trị cho nhiều ứng dụng. Tuy nhiên, điều quan trọng cần xem xét các hạn chế tiềm ẩn của nó, chẳng hạn như gradient biến mất/bùng phát và công suất thấp hơn, khi so sánh với các mô hình RNN phức tạp hơn.

LSTM là viết tắt của Long Short-Term Memory, là một loại mạng nơ-ron hồi quy (RNN) được thiết kế để học các phụ thuộc dài hạn trong dữ liệu tuần tự.LSTM hoạt động bằng cách sử dụng các tế bào nhớ để lưu trữ thông tin về trạng thái trước đó của chuỗi. Điều này cho phép LSTM ghi nhớ các sự kiện xảy ra trước đây trong chuỗi và sử dụng thông tin đó để đưa ra dự đoán hoặc phân loại chính xác hơn.

Lớp đầu vào: Lớp đầu vào nhận một tensor 2D có hình dạng (TIMESTEPS, NBR\_FEATURES), trong đó TIMESTEPS đại diện cho số bước thời gian trong chuỗi đầu vào và NBR\_FEATURES đại diện cho số tính năng cho mỗi bước thời gian.

Lớp tích chập: Lớp tích chập đóng vai trò là bộ trích xuất tính năng, trích xuất thông tin liên quan từ chuỗi đầu vào. Nó sử dụng một lớp tích chập 1D với 32 bộ lọc, kích thước hạt là 8 và bước là 1. Hàm kích hoạt 'relu' đưa ra tính phi tuyến tính, cho phép mô hình học các mẫu phức tạp. Chèn 'same' đảm bảo rằng chuỗi đầu ra có cùng độ dài với chuỗi đầu vào.

Lớp Bidirectional LSTM[9]: Hai lớp Bidirectional LSTM được sử dụng để nắm bắt các phụ thuộc dài hạn trong chuỗi. Lớp Bidirectional LSTM xử lý chuỗi theo cả hướng tiến và lùi, cho phép chúng nắm bắt thông tin ngữ cảnh từ cả hai phía của chuỗi. Mỗi lớp Bidirectional LSTM bao gồm TIMESTEPS đơn vị LSTM. Tham số return\_sequences được đặt thành True, chỉ ra rằng đầu ra của mỗi lớp Bidirectional LSTM là một chuỗi vectơ.

Lớp đầu ra: Lớp đầu ra bao gồm một đơn vị dày 1D với hàm kích hoạt 'sigmoid'. Đầu ra đại diện cho xác suất của một sự kiện cụ thể xảy ra, phù hợp cho các tác vụ phân loại nhị phân.

Nhìn chung, LSTM là một công cụ mạnh mẽ có thể được sử dụng cho nhiều ứng dụng dự đoán và phân loại liên quan đến dữ liệu tuần tự.

1.4 Thực nghiệm

1.4.1 Dữ liệu

Tập dữ liệu bao gồm khoảng 500 bản ghi dữ liệu gia tốc kế đeo ở cổ tay trong nhiều ngày được chú thích bằng hai loại sự kiện: khởi phát , bắt đầu giấc ngủ và thức dậy , kết thúc giấc ngủ.

Mặc dù nhật ký giấc ngủ vẫn là tiêu chuẩn vàng nhưng khi làm việc với dữ liệu gia tốc kế, chúng tôi coi giấc ngủ là khoảng thời gian không hoạt động dài nhất khi đang đeo đồng hồ. Đối với dữ liệu này, chúng tôi đã hướng dẫn người xếp hạng một số hướng dẫn cụ thể:

- Một lần ngủ phải kéo dài ít nhất 30 phút

- Một khoảng thời gian ngủ có thể bị gián đoạn bởi các hoạt động không quá 30 phút liên tục

- Không thể phát hiện cửa sổ ngủ trừ khi đồng hồ được cho là đã được đeo trong thời gian đó (được trình bày chi tiết bên dưới)

- Khoảng thời gian ngủ dài nhất trong đêm là khoảng thời gian duy nhất được ghi lại

- Nếu không có khoảng thời gian ngủ hợp lệ nào có thể nhận dạng được thì cả sự kiện bắt đầu và sự kiện đánh thức đều không được ghi lại cho đêm đó.

Các sự kiện về giấc ngủ không cần phải xếp chồng lên nhau trong ngày và do đó không có quy tắc cứng nhắc nào xác định số lượng có thể xảy ra trong một khoảng thời gian nhất định. Tuy nhiên, không nên chỉ định nhiều hơn một cửa sổ mỗi đêm. Ví dụ: một cá nhân có thể có thời gian ngủ từ 01h00–06h00 và 19h00–23h30 trong cùng một ngày dương lịch, mặc dù được chỉ định cho các đêm liên tiếp. Số đêm được ghi lại cho một loạt phim gần bằng số khoảng thời gian 24 giờ trong loạt phim đó.

Mặc dù mỗi chuỗi là một bản ghi liên tục nhưng có thể có những khoảng thời gian trong chuỗi khi thiết bị đo gia tốc bị tháo ra. Những khoảng thời gian này được xác định là những khoảng thời gian mà tín hiệu gia tốc kế có sự thay đổi nhỏ đáng ngờ trong một khoảng thời gian dài, điều này không thực tế đối với những người tham gia thông thường.

1.4.2 Xử lý dữ liệu

Tập dữ liệu bao gồm khoảng 500 bản ghi dữ liệu gia tốc kế đeo ở cổ tay trong nhiều ngày được chú thích bằng hai loại sự kiện: khởi phát , bắt đầu giấc ngủ và thức dậy , kết thúc giấc ngủ.

Mặc dù nhật ký giấc ngủ vẫn là tiêu chuẩn vàng nhưng khi làm việc với dữ liệu gia tốc kế, chúng tôi coi giấc ngủ là khoảng thời gian không hoạt động dài nhất khi đang đeo đồng hồ. Các bước tiền xử lý dữ liệu được thực hiện nhằm mục đích chuẩn hóa dữ liệu và cải thiện hiệu quả của mô hình phân loại. Các bước tiền xử lý bao gồm:

- Feature extraction: thực hiện quy trình chọn mẫu một số lượng chuỗi huấn luyện từ tập dữ liệu gốc, sau đó tạo ra một tập dữ liệu huấn luyện mới bằng cách hợp nhất thông tin từ hai DataFrame train\_series và train\_events dựa trên các cột chung. Điều này có thể giúp giảm kích thước của tập dữ liệu và tăng tốc quá trình huấn luyện trong môi trường máy học.

- Create new data: Tạo các mẫu : tạo ra một tập dữ liệu mẫu xung quanh các sự kiện "onset" trong tập dữ liệu huấn luyện (train\_merged). Quá trình này được thực hiện bằng cách thực hiện các bước sau:

Các bước để tạo tập dữ liệu mẫu như sau:

* Tìm tất cả các sự kiện "onset"/ "wakeup" trong tập dữ liệu huấn luyện.
* Đối với mỗi sự kiện "onset"/ "wakeup", tạo một cửa sổ dữ liệu có chiều dài 150 bước, bắt đầu từ 75 bước trước sự kiện "onset" và kết thúc 75 bước sau sự kiện "onset"/ "wakeup".
* Thêm một cột mới vào cửa sổ dữ liệu, có tên là "asleep".
* Gán giá trị 1 cho cột "asleep" cho tất cả các bước trong cửa sổ dữ liệu, từ sự kiện "onset" đến sự kiện "wakeup".
* Gán giá trị 0 cho cột "asleep" cho tất cả các bước còn lại trong cửa sổ dữ liệu.
* Tập dữ liệu mẫu sau đó có thể được sử dụng để huấn luyện mô hình học máy. Mô hình học máy sẽ học cách dự đoán trạng thái ngủ dựa trên dữ liệu từ các biến "enmo" và "anglez". Cột "asleep" trong các mẫu được sử dụng làm nhãn để mô hình có thể học cách dự đoán trạng thái ngủ.

- Thêm các đặc trưng mới và Nối các mẫu:

Tạo hai DataFrame df1 và df2, mỗi DataFrame chứa dữ liệu từ các sự kiện "onset" và "wakeup"

Tính toán các đặc trưng mới cho mỗi DataFrame. Các đặc trưng mới này bao gồm sự thay đổi giữa các giá trị liên tiếp và giá trị trung bình trượt.

Sự thay đổi giữa các giá trị liên tiếp (diff) cho phép mô hình học máy nắm bắt được xu hướng thay đổi của dữ liệu. Ví dụ, nếu giá trị "enmo" tăng đột ngột, mô hình có thể học được rằng đây là dấu hiệu của một sự thay đổi trạng thái ngủ.

Giá trị trung bình trượt (rolling mean) cho phép mô hình học máy nắm bắt được xu hướng trung bình của dữ liệu trong một khoảng thời gian nhất định. Ví dụ, nếu giá trị "anglez" trung bình tăng lên trong vòng 12 bước, mô hình có thể học được rằng đây là dấu hiệu của một sự thay đổi trạng thái ngủ.

Nối hai DataFrame thành một DataFrame lớn.Việc nối hai DataFrame thành một DataFrame lớn giúp mô hình học máy có thể học hỏi từ dữ liệu của cả hai sự kiện.

Điền các giá trị thiếu bằng giá trị liền kề.Việc điền các giá trị thiếu bằng giá trị liền kề giúp loại bỏ các giá trị NaN xuất hiện sau các phép toán diff và rolling mean. Giá trị NaN có thể gây ra sai lệch cho mô hình học máy.

In ra số lượng mẫu cho quá trình huấn luyện.Tổng số lượng mẫu cho quá trình huấn luyện được tính bằng cách cộng số lượng mẫu từ mỗi DataFrame.

Normalization: Dữ liệu đầu vào X chứa thông tin về các đặc trưng theo thời gian.Dữ liệu đầu ra y chứa nhãn cho mỗi bước thời gian trong mỗi mẫu. Nhãn có thể là 0 hoặc 1, đại diện cho trạng thái ngủ hoặc thức dậy.Mảng X được định hình lại thành một ba chiều, với chiều thứ nhất là số mẫu huấn luyện, chiều thứ hai là số bước thời gian, và chiều thứ ba là số đặc trưng. Mảng y được định hình lại thành một hai chiều, với chiều thứ nhất là số mẫu huấn luyện và chiều thứ hai là số bước thời gian.

1.4.3 Công nghệ sử dụng

|  |  |
| --- | --- |
| Ngôn ngữ | Python |
| Thư viện | Keras,Tensorflow |
| Môi trường | Kaggle GPU P100 |

**Bảng 1:** Công nghệ sử dụng

1.4.4 Cách đánh giá

Average Precision Score[10] là một phương thức đánh giá hiệu suất của mô hình phân loại, đặc biệt là trong trường hợp mô hình đang làm việc với dữ liệu không cân bằng, nơi số lượng mẫu thuộc một lớp có thể lớn hơn nhiều so với lớp còn lại. Đây là một phương pháp đánh giá chất lượng của mô hình dựa trên mức độ chính xác của việc xếp hạng các dự đoán.

Precision (pn)[11]: Là tỷ lệ số lượng dự đoán đúng tích cực (true positives) trên tổng số lượng dự đoán tích cực (true positives + false positives). Nó đo lường khả năng của mô hình trong việc không làm sai lạc một ví dụ tích cực.

Recall (rn)[12]: Là tỷ lệ số lượng dự đoán đúng tích cực trên tổng số lượng thực tế là tích cực (true positives + false negatives). Nó đo lường khả năng của mô hình trong việc bắt được tất cả các ví dụ tích cực.Bây giờ, chúng ta sẽ giải thích cách tính trung bình chính xác (average precision - AP) từ các điểm dự đoán:

Các bước thực hiện:

Giả sử có  gưỡng để tính precision và recall, với mỗi ngưỡng cho một cặp giá trị precision, recall là **,, = 1,2,…..**  Precision-Recall curve[13] được vẽ bằng cách vẽ từng điểm có toạ độ (**,)**  trên trục toạ độ và nối chúng với nhau. AP được xác định bằng:

Trong công thức, và là precision và recall tại ngưỡng thứ . Cách thực hiện này không sử dụng nội suy (interpolation) và khác biệt so với việc tính diện tích dưới đường cong precision-recall bằng quy tắc hình thang.

Điểm cao thường chỉ ra một mô hình có khả năng dự đoán chính xác và ổn định ở nhiều mức độ recall khác nhau, là một đánh giá quan trọng trong trường hợp dữ liệu không cân bằng.

Độ chính xác (Accuracy)[15]

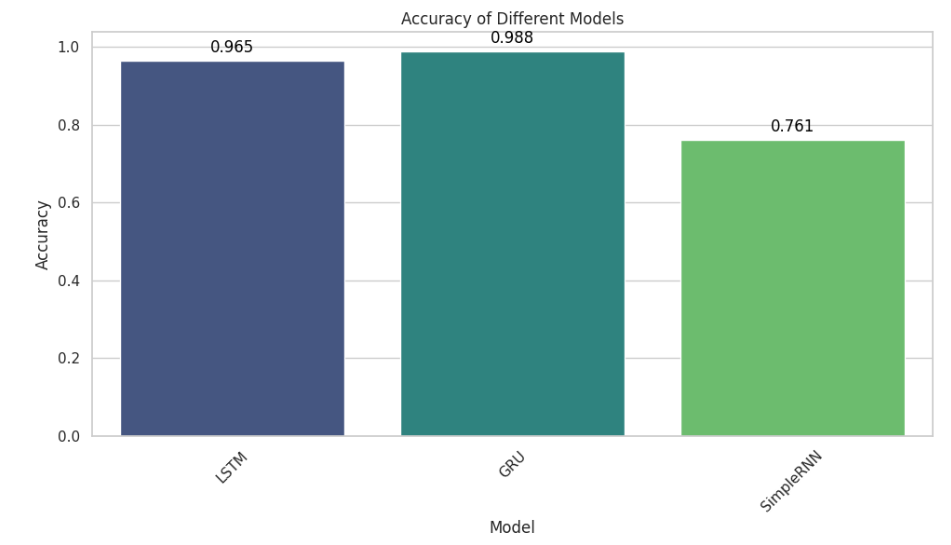
Độ chính xác được tính bằng cách so sánh số dự đoán đúng với tổng số mẫu trong tập kiểm thử.

# 1.5 Kết quả đạt được

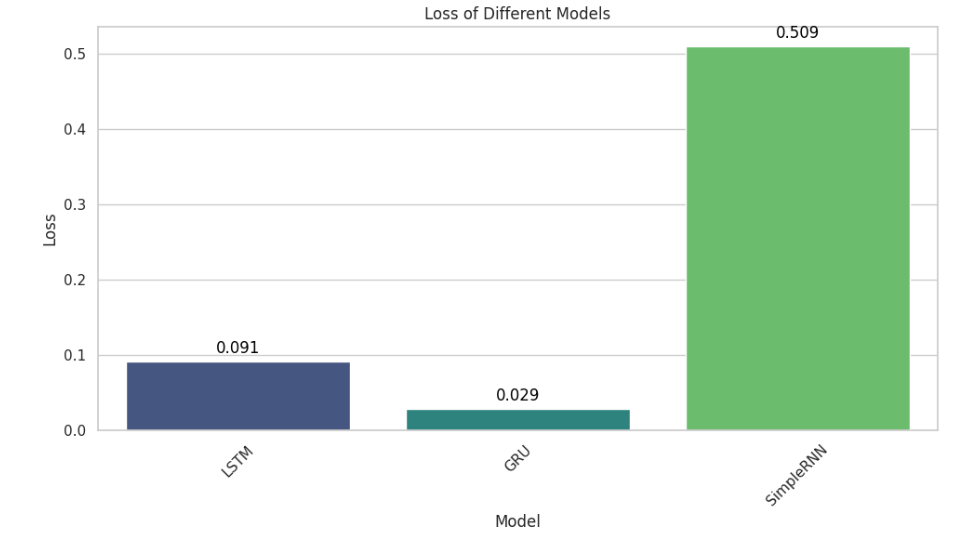
Đồ án này sử dụng bộ tối ưu hóa Adamax[14] với learning rate 1e-3. Batch size là 32. Chúng tôi huấn luyện các mô hình với 40 epoch, sử dụng các callback [] để lưu lại mô hình với các điểm checkpoint tốt nhất.

Kết quả

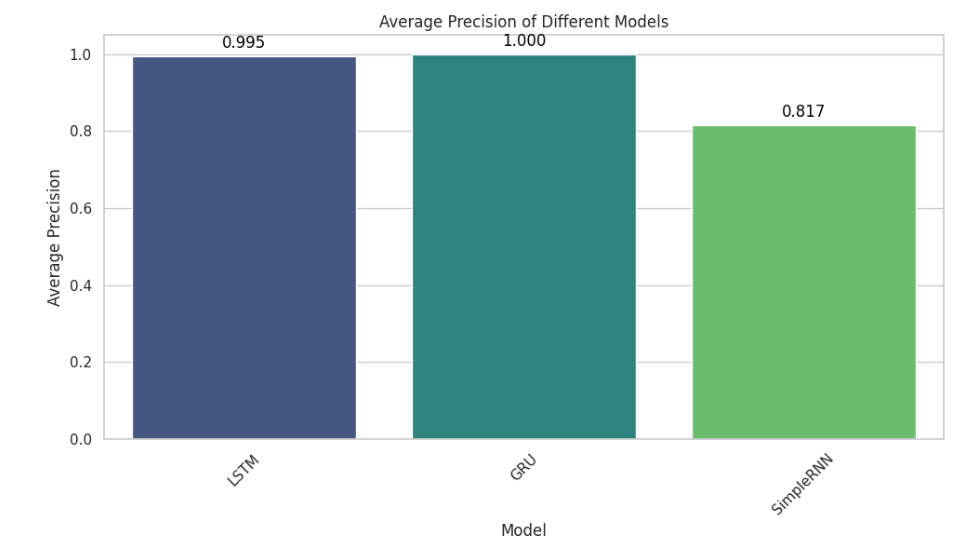
Dựa trên các phương pháp, tham số đã được nêu ra ở trên, chúng tôi tiến hành thực nghiệm để dự đoán trạng thái của giấc ngủ .



**Hình 5:** Accuracy giữa các mô hình



**Hình 6:** Loss giữa các mô hình



Hình 7: Average Precision Score giữa các mô hình

Thông qua các biểu độ trực quan hóa ở trên, ta thấy mô hình GRU mang lại kết quả tốt nhất. Do đó, chúng tôi tiến hành dự đoán một số mẫu dựa trên mô hình này với các đặc trưng anglez, enmo, anglez\_diff, enmo\_diff, anglez\_ma, enmo\_ma.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Series\_id | anglez | enmo | anglez\_diff | enmo\_diff | anglez\_ma | enmo\_ma |
| 038441c925bb | -80.01 | 0.01 | -9.99e-04 | 5.00e-04 | -80.01 | 0.01 |
| 0402a003dae9 | -11.72 | 0.07 | 5.85e+01 | 1.76e-02 | -60.14 | 0.05 |
| 03d92c9f6f8a | -88.22 | 0.00 | 0.00e+00 | 0.00e+00 | -88.22 | 0.00 |

Bảng 2: Các mẫu dự đoán

Và cho ra các kết quả như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Series\_id | Step | Event | Score |
| 038441c925bb | 77 | onset | 0.56 |
| 0402a003dae9 | 62 | onset | 0.50 |
| 03d92c9f6f8a | 143 | oset | 0.51 |
| 03d92c9f6f8a | 145 | wakeup | 0.48 |

**Bảng 3:** Kết quả thực nghiệm

Kết quả so sánh:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Loss | Average Precision Score |
| Simplernn | 0.761 | 0.509 | 0.817 |
| GRU | 0.988 | 0.029 | 1 |
| LSTM | 0.965 | 0.091 | 0.995 |

**Bảng 4:** Kết quả so sánh

Từ các kết quả so sánh trên có thể thấy mô hình GRU cho ra các kết quả tốt hơn so với các mô hình còn lại.

1.6 Kết luận

Về mặt lý thuyết, Đồ án đã tìm hiểu về các phương pháp giải quyết bài toán dự đoán được trạng thái ngủ , đồng thời Đồ án cũng đề xuất phương pháp học sâu để giải quyết bài toán này.

Kết quả thực nghiệm của nghiên cứu đã chứng minh tính hiệu quả và tiềm năng của phương pháp đề xuất trong bài toán dự đoán trạng thái giấc ngủ . Thực nghiệm trên các tập dữ liệu đa dạng đã chứng minh tính chính xác trong phương pháp của chúng tôi và các phương pháp khác.

Phương pháp làm này bị hạn chế trong dự báo dài hạn, vì các mô hình học sâu thường bị ảnh hưởng bởi vấn đề vanishing gradient hoặc exploding gradient bị thiếu khả năng giải thích và minh bạch, vì các mô hình học sâu thường có kiến trúc phức tạp và có nhiều tham số khi sử dụng các mạng neural hồi quy như LSTM hoặc GRU.

# LÀM VIỆC NHÓM

Cách thức làm việc nhóm: gặp nhau trực tiếp để trao đổi, tìm kiếm tài liệu hoặc họp online qua discord,qua các buổi thực hành trên lớp.

Phân chia công việc của các thành viên trong nhóm:

|  |  |  |
| --- | --- | --- |
| Thành viên | Nhiệm vụ | Mức độ hoàn thành |
| Nguyễn Quang Bảo | 1.3 Phương pháp giải quyết bài toán  1.4 Thực nghiệm  1.5 Kết quả đạt được | 100% |
| Nguyễn Xuân Giang | 1.1 Giới thiệu về bài toán 1.2 Phân tích yêu cầu của bài toán 1.6 Kết luận | 100% |

Tổng số lần gặp nhau:

Trực tiếp: 5 lần

Online: 10 lần

Tổng thời gian gặp nhau: Khoảng 25 giờ

# TÀI LIỆU THAM KHẢO

1. Sundararajan et al. (2020) trong bài báo “*Sleep classification from wrist-worn accelerometer data using random forests”* Chen et al. (2021) trong bài báo *Three-level Sleep Stage Classification Based on Wrist-worn Accelerometer Data Using Recurrent Neural Networks*
2. Chen et al. (2021) trong bài báo “*Three-level Sleep Stage Classification Based on Wrist-worn Accelerometer Data Using Recurrent Neural Networks”*
3. Kortelainen et al. (2020) trong bài báo “*Estimation of respiration rate and sleeping position using a wearable accelerometer”*
4. Koutnik, Jan, et al. "*A clockwork rnn.*" International conference on machine learning. PMLR, 2014.
5. He, Kaiming, et al. "*Mask r-cnn.*" Proceedings of the IEEE international conference on computer vision. 2017.
6. Yu, Yong, et al. "*A review of recurrent neural networks: LSTM cells and network architectures.*" Neural computation 31.7 (2019): 1235-1270.
7. Prabowo, Yulius Denny, et al. "*Lstm and simple rnn comparison in the problem of sequence to sequence on conversation data using bahasa indonesia*." 2018 Indonesian association for pattern recognition international conference (INAPR). IEEE, 2018.
8. Saha, Prasenjit, and Scott Tremaine. "*Long-term planetary integration with individual time steps.*" arXiv preprint astro-ph/9403057 (1994).
9. Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. "*Hybrid speech recognition with deep bidirectional LSTM.*" 2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013.
10. Yue, Yisong, et al. "*A support vector method for optimizing average precision.*" Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 2007.
11. König, Inke R., et al. "*What is precision medicine?*." European respiratory journal 50.4 (2017).
12. Bell, Gordon. Total recall. Gramedia Pustaka Utama, 2013.
13. Davis, Jesse, and Mark Goadrich. "*The relationship between Precision-Recall and ROC curves.*" Proceedings of the 23rd international conference on Machine learning. 2006.
14. Llugsi, Ricardo, et al. "*Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito.*" 2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM). IEEE, 2021.
15. Johansson, Magnus, Martin Lovmar, and Måns Ehrenberg. "*Rate and accuracy of bacterial protein synthesis revisited.*" Current opinion in microbiology 11.2 (2008): 141-147.

TỰ ĐÁNH GIÁ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Nội dung | Điểm chuẩn | Tự chấm | Ghi chú |
| 1  (8.5) | 1.1 Giới thiệu về bài toán | 0.5 | 0.5 |  |
| 1.2 Phân tích yêu cầu của bài toán | 1 | 1 |  |
| 1.3 Phương pháp giải quyết bài toán | 1.5 | 1.5 |  |
| 1.4 Thực nghiệm | 4 | 3.5 |  |
| **1.5 Kết quả đạt được** | 1 | 0.5 |  |
| 1.6 Kết luận | 0.5 | 0.5 |  |
| 2  (1) | Báo cáo (chú ý các chú ý 2,3,4,6 ở trang trước, nếu sai sẽ bị trừ điểm nặng) | 1đ | 0.5 |  |
| 3  (0.5) | Điểm nhóm (chú ý trả lời các câu hỏi trong mục làm việc nhóm) | 0.5đ | 0.25 |  |
| Tổng điểm | | | 8 |  |