HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1

****

TÌM HIỂU DEEP LEARNING TRONG DỰ ĐOÁN BỆNH TIỂU ĐƯỜNG

MÔN HỌC: KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM

Sinh viên: **Nguyễn Mạnh Hùng**

Mã sinh viên: **B21DCCN412**

Giảng viên: **PGS.TS. Trần Đình Quế**

Hà Nội, 2025

MỤC LỤC

1. Giới thiệu

Deep Learning (DL) hay còn gọi là học sâu, là một nhánh của Machine Learning (ML), tập trung vào việc sử dụng các mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN) có nhiều tầng (deep neural networks). Mô hình DL có khả năng tự động trích xuất đặc trưng từ dữ liệu, giúp cải thiện hiệu suất trong các bài toán phức tạp mà các phương pháp ML truyền thống khó xử lý.

Mô hình Deep Learning hoạt động dựa trên nguyên lý mô phỏng cách thức hoạt động của bộ não con người, trong đó mỗi nơ-ron nhân tạo (neuron) nhận đầu vào, áp dụng trọng số và hàm kích hoạt để tạo ra đầu ra, sau đó truyền thông tin qua các tầng ẩn (hidden layers) trước khi tạo ra kết quả cuối cùng. Quá trình huấn luyện được thực hiện bằng thuật toán lan truyền ngược (Backpropagation) và tối ưu hóa bằng các phương pháp như Gradient Descent.

Machine Learning và Deep Learning đều hướng đến mục tiêu giúp máy tính có khả năng học hỏi từ dữ liệu để đưa ra dự đoán hoặc quyết định. Tuy nhiên, chúng có một số điểm khác biệt quan trọng:

Tiêu chí	**Machine Learning**	**Deep Learning**
Cách trích xuất đặc trưng	Cần chuyên gia thiết kế đặc trưng thủ công	Tự động trích xuất đặc trưng từ dữ liệu
Kiến trúc mô hình	Các thuật toán như Decision Tree, SVM, kNN, Logistic Regression	Mạng nơ-ron nhân tạo nhiều tầng (DNN, CNN, RNN)
Dữ liệu yêu cầu	Cần ít dữ liệu hơn	Cần lượng dữ liệu lớn để đạt hiệu suất tốt
Tính toán	Ít phức tạp hơn, có thể chạy trên CPU	Yêu cầu tài nguyên cao, thường cần GPU để tăng tốc
Ứng dụng	Phù hợp với bài toán nhỏ, dữ liệu có cấu trúc	Phù hợp với bài toán phức tạp như nhận diện hình ảnh, xử lý ngôn ngữ tự nhiên

Nhìn chung, DL vượt trội hơn ML khi xử lý dữ liệu phi cấu trúc, đặc biệt là hình ảnh, âm thanh và văn bản. Tuy nhiên, nhược điểm của DL là yêu cầu phần cứng mạnh mẽ và lượng dữ liệu lớn để đạt hiệu quả tốt.

Deep Learning đang có những bước tiến lớn trong ngành y tế nhờ khả năng phân tích dữ liệu nhanh chóng và chính xác. Một số ứng dụng tiêu biểu của DL trong y học bao gồm:

- **Chẩn đoán hình ảnh y khoa:** DL được sử dụng để phân tích ảnh chụp X-quang, MRI, CT scan để phát hiện ung thư, bệnh về phổi, tổn thương thần kinh, v.v.
- **Dự đoán bệnh tật:** Các mô hình DL có thể phân tích dữ liệu y tế (chỉ số xét nghiệm, lịch sử bệnh án) để dự đoán nguy cơ mắc bệnh như tiểu đường, tim mạch.
- **Xử lý ngôn ngữ y khoa:** DL giúp trích xuất thông tin từ tài liệu y tế, tự động hóa việc ghi chú bệnh án và hỗ trợ bác sĩ trong nghiên cứu.
- **Phát triển thuốc:** Mô hình DL có thể phân tích hàng triệu hợp chất hóa học để tìm ra loại thuốc tiềm năng mới.

Trong bài toán dự đoán bệnh tiểu đường, DL có thể học từ dữ liệu bệnh nhân để tìm ra các mô hình tiềm ẩn giúp phân biệt giữa người khỏe mạnh và người có nguy cơ mắc bệnh. Việc áp dụng DL không chỉ giúp tăng độ chính xác của chẩn đoán mà còn hỗ trợ bác sĩ đưa ra quyết định nhanh chóng hơn.

2. Kiến trúc của Deep learning

2.1. Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN)

Mạng nơ-ron nhân tạo (ANN) là nền tảng của Deep Learning, mô phỏng hoạt động của các nơ-ron sinh học trong não bộ con người. ANN gồm nhiều lớp kết nối với nhau, mỗi lớp chứa các nơ-ron nhân tạo thực hiện nhiệm vụ tính toán và truyền dữ liệu.

Một mạng nơ-ron cơ bản bao gồm ba thành phần chính:

- **Nơ-ron (Neuron):** Là đơn vị cơ bản xử lý thông tin, mỗi nơ-ron nhận đầu vào, áp dụng trọng số và hàm kích hoạt để tạo ra đầu ra.
- **Trọng số (Weights):** Xác định mức độ ảnh hưởng của mỗi đầu vào đến kết quả đầu ra. Trong quá trình học, mô hình điều chỉnh trọng số để tối ưu hóa kết quả.
- **Hàm kích hoạt (Activation Function):** Giúp mô hình học các quan hệ phi tuyến tính giữa đầu vào và đầu ra.

Quá trình xử lý dữ liệu trong một mạng nơ-ron diễn ra như sau:

1. **Nhận đầu vào:** Mỗi nơ-ron nhận một tập hợp giá trị đầu vào (x_1, x_2, ..., x_n).
2. **Tính toán tổng có trọng số:** Giá trị đầu vào được nhân với trọng số tương ứng, sau đó cộng dồn lại với bias (b). z=w1x1+w2x2+...+wnxn+bz = w_1x_1 + w_2x_2 + ... + w_nx_n + bz=w1x1+w2x2+...+wnxn+b
3. **Áp dụng hàm kích hoạt:** Kết quả zzz được đưa qua hàm kích hoạt để tạo ra đầu ra phi tuyến tính.
4. **Truyền thông tin:** Đầu ra của một nơ-ron sẽ trở thành đầu vào của các nơ-ron ở tầng kế tiếp.

2.2. Mạng nơ-ron nhiều lớp (Deep Neural Networks - DNN)

Khi ANN có nhiều hơn một tầng ẩn (hidden layer), ta gọi đó là mạng nơ-ron nhiều lớp (Deep Neural Networks - DNN). Các tầng ẩn đóng vai trò quan trọng trong việc trích xuất đặc trưng từ dữ liệu, giúp mô hình học các biểu diễn phức tạp hơn.

- **Tầng đầu vào (Input Layer):** Nhận dữ liệu thô từ tập huấn luyện.
- **Tầng ẩn (Hidden Layers):** Mỗi tầng xử lý một mức độ trừu tượng khác nhau của dữ liệu, từ đặc trưng đơn giản đến phức tạp.
- **Tầng đầu ra (Output Layer):** Dự đoán kết quả cuối cùng, có thể là một hoặc nhiều giá trị (nhị phân hoặc đa lớp).

Lan truyền ngược (Backpropagation) là thuật toán tối ưu hóa trọng số trong mạng nơ-ron dựa trên sai số giữa đầu ra thực tế và đầu ra dự đoán.

Quá trình huấn luyện diễn ra qua các bước sau:

1. **Lan truyền tiến (Forward Propagation):** Dữ liệu đầu vào đi qua các tầng của mạng, tính toán đầu ra dự đoán.
2. **Tính toán lỗi (Loss Calculation):** Sai số được đo lường bằng một hàm mất mát (Loss Function), ví dụ Binary Cross-Entropy cho bài toán nhị phân.
3. **Lan truyền ngược (Backward Propagation):** Mô hình điều chỉnh trọng số bằng cách sử dụng đạo hàm của hàm mất mát theo từng trọng số.
4. **Cập nhật trọng số (Gradient Descent):** Sử dụng thuật toán tối ưu như Gradient Descent hoặc Adam để cập nhật trọng số và giảm lỗi.

2.3. Các hàm kích hoạt phổ biến

Hàm kích hoạt giúp mạng nơ-ron học được các mối quan hệ phi tuyến tính. Dưới đây là một số hàm phổ biến:

1. ReLU (Rectified Linear Unit)

$$f(x) = \max(0, x)$$

Ưu điểm:

- Giải quyết được vấn đề biến mất gradient (vanishing gradient) của Sigmoid/Tanh.
- Hiệu quả tính toán cao, đơn giản.

Nhược điểm: Gặp vấn đề "Dying ReLU" khi giá trị âm luôn bằng 0, khiến một số nơ-ron ngừng cập nhật.

2. Sigmoid

$$f(x) = \frac{1}{1 + e^{-x}}$$

Ưu điểm: Đầu ra nằm trong khoảng (0,1), thích hợp cho bài toán phân loại nhị phân.

Nhược điểm: Gặp vấn đề biến mất gradient, làm chậm quá trình huấn luyện.

3. Tanh (Hyperbolic Tangent)

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Ưu điểm: Giống Sigmoid nhưng đầu ra nằm trong khoảng (-1,1), giúp mô hình hội tụ nhanh hơn.

Nhược điểm: Vẫn gặp vấn đề biến mất gradient ở đầu vào lớn hoặc nhỏ.

Trong mô hình của bài toán dự đoán bệnh tiểu đường, chúng ta đã sử dụng **ReLU** cho các tầng ẩn và **Sigmoid** cho đầu ra để đưa ra dự đoán nhị phân.

3. **Deep** Learning trong bài toán **dự đoán bệnh tiểu đường (1.5 trang)**

3.1. Dữ liệu Pima Indians Diabetes Dataset

Bộ dữ liệu **Pima Indians Diabetes** là một tập dữ liệu y tế chứa thông tin về bệnh tiểu đường ở phụ nữ người Mỹ gốc Pima. Dữ liệu này được thu thập bởi Viện Tiểu đường và Bệnh Tiêu hóa & Thận Quốc gia Hoa Kỳ, và được sử dụng phổ biến trong các nghiên cứu về học máy và trí tuệ nhân tạo.

Bộ dữ liệu có **768 mẫu**, mỗi mẫu đại diện cho một bệnh nhân với **8 đặc trưng đầu vào** và một nhãn đầu ra (0: Không mắc bệnh, 1: Mắc bệnh).

Bảng dưới đây liệt kê các đặc trưng của bộ dữ liệu:

Tên đặc trưng	Ý nghĩa
Pregnancies	Số lần mang thai
Glucose	Nồng độ glucose trong máu
Blood Pressure	Huyết áp (mm Hg)
Skin Thickness	Độ dày nếp gấp da
Insulin	Mức insulin trong máu
BMI	Chỉ số khối cơ thể (Body Mass Index)
Diabetes Pedigree Function	Chỉ số di truyền bệnh tiểu đường
Age	Tuổi của bệnh nhân

3.2. Xây dựng mô hình Deep Learning

Mô hình Deep Learning được thiết kế với nhiều tầng ẩn để học các đặc trưng từ dữ liệu và cải thiện độ chính xác trong việc dự đoán bệnh tiểu đường.

Kiến trúc mô hình

Mô hình sử dụng **Mạng nơ-ron nhiều lớp (DNN)** với kiến trúc như sau:

- **Tầng đầu vào:** Kích thước đầu vào là số lượng đặc trưng của dữ liệu (8).
- **Các tầng ẩn:**
    - **256 neuron**, activation = ReLU, dropout = 0.4
    - **128 neuron**, activation = ReLU, dropout = 0.3

    - **64 neuron**, activation = ReLU, dropout = 0.3
    - **32 neuron**, activation = ReLU
    - **16 neuron**, activation = ReLU
- **Tầng đầu ra:** 1 neuron, activation = Sigmoid (dự đoán nhị phân).

Hàm kích hoạt (Activation Function)

- **ReLU (Rectified Linear Unit)** được sử dụng cho các tầng ẩn giúp tránh vấn đề vanishing gradient.
- **Sigmoid** được sử dụng ở tầng đầu ra để đưa ra dự đoán có giá trị trong khoảng (0,1), phù hợp cho bài toán phân loại nhị phân.

Hàm mất mát (Loss Function) và thuật toán tối ưu (Optimizer)

- **Binary Cross-Entropy** được sử dụng làm hàm mất mát do đây là bài toán phân loại nhị phân.
- **Adam Optimizer** được sử dụng để cập nhật trọng số, giúp mô hình hội tụ nhanh hơn và đạt hiệu suất tốt hơn.

Mô hình được huấn luyện trong **50 epochs** với batch size = **16**, sử dụng tập huấn luyện (**X_train, y_train**) và kiểm tra hiệu suất trên tập kiểm tra (**X_test, y_test**).

3.3. So sánh với các mô hình truyền thống

Sau khi huấn luyện mô hình Deep Learning, ta so sánh độ chính xác của nó với các mô hình truyền thống khác như Logistic Regression, SVM và kNN.

Mô hình	**Độ chính xác**
k-Nearest Neighbors (kNN)	0.7467
SVM (Linear Kernel)	0.7632
SVM (RBF Kernel)	0.7662
Logistic Regression	0.7597
Deep Learning (DNN)	**0.7821**

Nhận xét:

- **Mô hình Deep Learning có độ chính xác cao nhất (78.21%)**, vượt trội hơn các mô hình truyền thống.
- **SVM với RBF Kernel cũng đạt hiệu suất khá tốt (76.62%)**, do kernel phi tuyến giúp mô hình học các quan hệ phức tạp trong dữ liệu.

- **Logistic Regression và kNN có độ chính xác thấp hơn**, do các phương pháp này gặp khó khăn khi dữ liệu có tính phi tuyến.

Ưu điểm của Deep Learning trong xử lý dữ liệu phi tuyến

- **Tự động trích xuất đặc trưng:** Không cần chọn đặc trưng thủ công như các mô hình ML truyền thống.
- **Xử lý tốt dữ liệu phi tuyến:** Mạng nơ-ron có thể học các quan hệ phức tạp trong dữ liệu.
- **Khả năng tổng quát hóa tốt:** Với số lượng lớn dữ liệu, mô hình có thể dự đoán chính xác hơn.

4. Những thách thức và cải tiến mô hình Deep Learning

4.1. Overfitting và cách khắc phục

Overfitting xảy ra khi mô hình học quá kỹ dữ liệu huấn luyện và không thể tổng quát hóa tốt trên dữ liệu mới. Điều này thường xảy ra khi mô hình quá phức tạp so với lượng dữ liệu hiện có.

1. **Regularization (L1, L2)**
    - **L1 Regularization (Lasso):** Áp dụng hình phạt lên trọng số, giúp loại bỏ các trọng số nhỏ, làm cho mô hình đơn giản hơn.
    - **L2 Regularization (Ridge):** Giữ cho trọng số nhỏ hơn, giảm nguy cơ overfitting.
    - Trong Keras, có thể sử dụng kernel_regularizer=l2(0.01) trong các lớp Dense.
2. **Dropout Layers**
    - Dropout là một kỹ thuật tắt ngẫu nhiên một số neuron trong quá trình huấn luyện, giúp mô hình không quá phụ thuộc vào một số đặc trưng cụ thể.
    - Trong mô hình của chúng ta, đã áp dụng dropout với tỉ lệ 0.3 - 0.4 ở các tầng ẩn.

4.2. Tối ưu hiệu suất mô hình

Điều chỉnh số lượng neuron, tầng ẩn

- Quá nhiều neuron có thể dẫn đến overfitting, trong khi quá ít neuron có thể khiến mô hình không học được đặc trưng quan trọng.
- Cần thử nghiệm với số lượng tầng và neuron khác nhau để tìm kiến trúc tối ưu.

Learning Rate, Batch Size, và Số Epoch

- **Learning Rate:** Nếu quá lớn, mô hình có thể bỏ qua điểm tối ưu; nếu quá nhỏ, mô hình học rất chậm.
    - Thường sử dụng các giá trị như 0.001, 0.0005.
- **Batch Size:** Ảnh hưởng đến tốc độ huấn luyện và độ ổn định của mô hình.
    - Giá trị nhỏ (16, 32) giúp mô hình linh hoạt hơn, nhưng thời gian huấn luyện dài hơn.
    - Giá trị lớn (128, 256) giúp huấn luyện nhanh hơn nhưng có thể bỏ qua các đặc trưng nhỏ.
- **Số Epoch:** Nếu quá ít, mô hình chưa học đủ; nếu quá nhiều, mô hình dễ bị overfitting.
    - Cần sử dụng Early Stopping để dừng huấn luyện khi mô hình không còn cải thiện.

4.3. Hướng phát triển trong tương lai

Sử dụng mô hình tiên tiến hơn

- **Convolutional Neural Networks (CNN):**
    - Thích hợp hơn nếu có hình ảnh y khoa như ảnh siêu âm, X-ray.
- **Recurrent Neural Networks (RNN):**
    - Có thể hữu ích nếu mở rộng bài toán sang phân tích dữ liệu thời gian (ví dụ: mức đường huyết theo thời gian).

Thu thập thêm dữ liệu để cải thiện độ chính xác

- Dữ liệu càng lớn, mô hình càng có khả năng tổng quát hóa tốt hơn.
- Cần thu thập dữ liệu từ nhiều nguồn hơn để đảm bảo tính đa dạng và độ tin cậy của mô hình.

5. Kết luận

Deep Learning đã chứng minh khả năng mạnh mẽ trong việc phân tích và xử lý dữ liệu y tế. Trong bài toán dự đoán bệnh tiểu đường, mô hình mạng nơ-ron nhiều lớp (DNN) có thể học được các đặc trưng quan trọng từ dữ liệu đầu vào và cho kết quả chính xác hơn so với nhiều mô hình truyền thống. Nhờ khả năng tự động trích xuất đặc trưng, Deep Learning giúp phát hiện mối quan hệ phức tạp giữa các yếu tố như chỉ số đường huyết, huyết áp, BMI, và tuổi tác để đưa ra dự đoán.

Mô hình Deep Learning với 6 tầng ẩn đã đạt độ chính xác **0.7338**, một kết quả khá tốt so với các mô hình truyền thống như kNN, SVM hay Logistic Regression. Dù không

đạt độ chính xác cao nhất trong các mô hình thử nghiệm, nhưng mô hình Deep Learning có khả năng mở rộng và tối ưu hóa hơn nữa khi có thêm dữ liệu và điều chỉnh tham số phù hợp.

Một số điểm nổi bật của mô hình Deep Learning trong bài toán này:

- Sử dụng **ReLU** giúp tăng khả năng học các đặc trưng phi tuyến tính.
- Sử dụng **Dropout** để giảm overfitting.
- Dùng **hàm loss binary_crossentropy** phù hợp cho bài toán phân loại nhị phân.
- **Adam Optimizer** giúp tối ưu hóa quá trình học của mô hình.

Mặc dù mô hình đã đạt kết quả khả quan, vẫn còn nhiều hướng để cải thiện:

- **Tăng cường dữ liệu:** Thu thập thêm dữ liệu để mô hình học tốt hơn, đặc biệt là từ nhiều nhóm dân cư khác nhau để tăng tính tổng quát.
- **Thử nghiệm các mô hình tiên tiến hơn:**
    - **CNN:** Nếu có dữ liệu hình ảnh, CNN có thể giúp phân tích hình ảnh y khoa liên quan đến bệnh tiểu đường.
    - **RNN hoặc LSTM:** Nếu mở rộng sang dữ liệu thời gian, các mô hình này có thể giúp dự đoán xu hướng đường huyết theo thời gian.
    - **Transformer-based models:** Áp dụng mô hình tiên tiến hơn để trích xuất đặc trưng từ dữ liệu y tế.
- **Tối ưu mô hình:**
    - Điều chỉnh số neuron, số tầng ẩn để tăng độ chính xác.
    - Thử nghiệm với các kỹ thuật regularization khác như Batch Normalization, Early Stopping.
    - Dùng hyperparameter tuning để tìm cấu hình tốt nhất.

Deep Learning là một công cụ mạnh mẽ trong lĩnh vực y tế, giúp nâng cao khả năng chẩn đoán bệnh tật thông qua dữ liệu. Với sự phát triển của công nghệ và lượng dữ liệu ngày càng tăng, Deep Learning sẽ tiếp tục đóng vai trò quan trọng trong việc hỗ trợ các chuyên gia y tế đưa ra quyết định chính xác hơn.