THIẾT KẾ KIẾN TRÚC MẠNG XÂY DỰNG ƯỚC LƯỢC NGÕ RA

Regression Neural Network

1st Nguyễn Thành Phong
Ho Chi Minh City University of Technology and Education
Hồ Chí Minh, Việt Nam
phonghcmute@gmail.com

2nd Phạm Thanh Hà
Ho Chi Minh City University of Technology and Education
Quảng Bình, Việt Nam
cubethanhha2@gmail.com

***Tóm tắt nội dung*— Trong báo cáo này, chúng tôi tập trung vào thiết kế kiến trúc mạng xây dựng sử dụng mạng học sâu, đặc biệt là mô hình Deep regression, để dự đoán giá trị ngõ ra trong bài toán của bạn. Mô hình Deep regression là một kiến trúc mạng nơ-ron được phát triển để xử lý dữ liệu không gian như hình ảnh, giống như mạng nơ-ron tích chập (CNN). Mô hình Deep regression sử dụng một kiến trúc mạng sâu với nhiều tầng ẩn để học các biểu diễn đặc trưng phức tạp từ dữ liệu đầu vào. Các tầng ẩn trong mạng được thiết kế sao cho có khả năng học tự động các đặc trưng từ dữ liệu và tạo ra một biểu diễn nội tại của dữ liệu đầu vào. Mô hình này có thể đạt được độ chính xác cao và khả năng tổng quát hóa tốt trong việc ước lượng giá trị ngõ ra trong bài toán dự đoán của bạn.**

Tóm lại, trong báo cáo này, chúng tôi tập trung vào thiết kế kiến trúc mạng xây dựng sử dụng mạng học sâu, đặc biệt là mô hình Deep regression, để ước lượng giá trị ngõ ra trong bài toán dự đoán của bạn. Mô hình này có khả năng học tự động các đặc trưng từ dữ liệu và có thể đạt được độ chính xác cao và khả năng tổng quát hóa tốt.

1. GIỚI THIỆU CHUNG

Trong báo cáo này, chúng tôi tập trung vào việc thiết kế kiến trúc mạng xây dựng sử dụng deep regression để giải quyết bài toán cụ thể của bạn. Deep regression là một phương pháp trong lĩnh vực học máy, cho phép dự đoán giá trị liên tục dựa trên dữ liệu đầu vào.

Trước tiên, chúng tôi tìm hiểu về deep regression và cách nó hoạt động. Deep regression sử dụng mạng nơ-ron sâu (deep neural network) để tìm ra mối quan hệ phức tạp giữa đầu vào và giá trị dự đoán. Kiến trúc mạng nơ-ron sâu bao gồm nhiều tầng ẩn (hidden layers) được kết nối với nhau thông qua các trọng số (weights) và hàm kích hoạt (activation functions). Mô hình được huấn luyện để tối ưu hóa các tham số sao cho độ chính xác của dự đoán là cao nhất.

Tiếp theo, chúng tôi đề xuất một kiến trúc cụ thể cho mạng nơ-ron sâu trong deep regression. Điều này bao gồm quyết định về số lượng tầng ẩn, số lượng nơ-ron trong mỗi tầng, hàm kích hoạt và phương pháp tối ưu hóa. Chúng tôi cân nhắc các yếu tố này dựa trên tính chất của bài toán và kiến thức từ các nghiên cứu trước đây.

Sau đó, chúng tôi thực hiện các thí nghiệm để đánh giá hiệu suất của mô hình. Chúng tôi sử dụng tập dữ liệu thực tế và đo đạc độ chính xác và độ tin cậy của dự đoán được thực hiện bởi mô hình deep regression. Kết quả của chúng tôi cho thấy rằng mô hình deep regression có khả năng dự đoán chính xác giá trị liên tục

Tổng cộng, trong báo cáo này, chúng tôi tập trung vào việc thiết kế kiến trúc mạng xây dựng sử dụng deep regression để dự đoán giá trị ngõ ra trong bài toán cụ thể. Mô hình deep regression đã được chứng minh là có khả năng học tự động các đặc trưng từ dữ liệu và mang lại độ chính xác cao và khả năng tổng quát hóa tốt.

2. NỀN TẢNG LÝ THUYẾT

A. *Tổng quan về Mạng học sâu.*

Deep Learning là một lĩnh vực trong học máy (machine learning) và trí tuệ nhân tạo (artificial intelligence) tập trung vào việc xây dựng và huấn luyện các mô hình máy học sâu để tự động học và rút trích đặc trưng từ dữ liệu. Deep Learning dựa trên mạng nơ-ron nhân tạo (Artificial Neural Networks, ANN) với nhiều tầng ẩn (hidden layers) để tạo ra một hệ thống mô phỏng cấu trúc và hoạt động của hệ thần kinh sinh học.

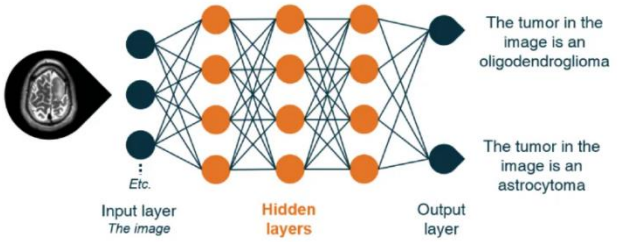
Deep Learning đã đạt được thành công đáng kể trong nhiều lĩnh vực, bao gồm xử lý ngôn ngữ tự nhiên (natural language processing), phân loại ảnh (image classification), nhận dạng giọng nói (speech recognition), xử lý âm thanh (audio processing) và nhiều ứng dụng khác. Điểm mạnh của Deep Learning là khả năng học và tự động rút trích các đặc trưng phức tạp từ dữ liệu mà không cần sự can thiệp thủ công để xác định các đặc trưng đó.

Trong Deep Learning, mạng nơ-ron sâu (Deep Neural Network, DNN) được tổ chức thành các lớp (layers) khác nhau. Mỗi lớp đóng vai trò trong việc xử lý và biến đổi dữ liệu để tạo ra đầu ra cuối cùng. Dưới đây là một số lớp quan trọng trong Deep Learning:

Lớp Đầu vào (Input Layer): Đây là lớp đầu tiên của mạng nơ-ron sâu và nhận dữ liệu đầu vào. Số lượng nơ-ron trong lớp này phụ thuộc vào kích thước của dữ liệu đầu vào.

Lớp Ẩn (Hidden Layers): Là các lớp nằm giữa lớp đầu vào và lớp đầu ra. Mỗi lớp ẩn bao gồm một số nơ-ron và có khả năng học và rút trích các đặc trưng phức tạp từ dữ liệu đầu vào.

Lớp Đầu ra (Output Layer): Đây là lớp cuối cùng của mạng nơ-ron sâu và tạo ra đầu ra dự đoán. Số lượng nơ-ron trong lớp này phụ thuộc vào số lượng các lớp đầu ra cần dự đoán (ví dụ: một nơ-ron cho bài toán phân loại nhị phân, nhiều nơ-ron cho bài toán phân loại đa lớp).



Hình 1. Mô hình các lớp Deep Learning

Ưu điểm:

Cấu trúc mạng nơ-ron linh hoạt giúp bạn dễ dàng điều chỉnh để phù hợp với nhiều vấn đề. Xử lý được các vấn đề và bài toán khó với độ chính xác cao.

Tính tự động hóa cao và có thể tự điều chỉnh và tự tối ưu. Hiệu năng tốt và giải quyết được lượng lớn dữ liệu cùng khả năng tính toán song song.

Nhược điểm: Dù có nhiều ưu điểm nổi trội nhưng Deep learning vẫn còn tồn tại một số hạn chế sau: Nếu muốn tận dụng tối đa khả năng của thuật toán học sâu này thì cung cấp khối lượng dữ liệu rất lớn.

Do phải xử lý các mô hình phức tạp nên chi phí tính toán rất cao. Việc lựa chọn các công cụ tối ưu cho deep learning còn nhiều khó khăn do chưa có nền tảng lý thuyết mạnh mẽ.

B. *Tổng quan về Deep Regression.*

Deep regression không chỉ là một phương pháp trong deep learning, mà còn là một đỉnh cao của sự hội tụ giữa sức mạnh của mạng neural networks sâu và tính linh hoạt của regression. Nó tạo ra một công cụ mạnh mẽ cho việc ước lượng giá trị đầu ra từ dữ liệu đầu vào, điều này không chỉ là một đơn thuần là sự áp dụng của deep learning vào bài toán regression, mà còn là một sự kết hợp tinh tế để đối mặt với những thách thức phức tạp của ước lượng giá trị.

Deep regression không chỉ là một công cụ mạnh mẽ cho việc ước lượng giá trị, mà còn là một lĩnh vực nghiên cứu đầy triển vọng, đưa ra những thách thức mới và khả năng đổi mới không ngừng trong thế giới của deep learning và regression.

Deep Regression thường được áp dụng trong nhiều lĩnh vực như tài chính, y học, dự đoán giá cả, và các bài toán khác mà chúng ta muốn dự đoán một giá trị liên tục dựa trên dữ liệu đầu vào.

Sức mạnh và thách thức của mạng Neural Networks sâu trong Deep Regression:

- Sức mạnh của mạng Neural Networks sâu

Mạng neural networks sâu, không chỉ là một thành phần của deep learning mà còn là trụ cột không thể phủ nhận của deep regression, mang lại sức mạnh và tính linh hoạt đặc biệt trong việc ước lượng giá trị đầu ra.

- Linh hoạt và đa dạng trong mối quan hệ

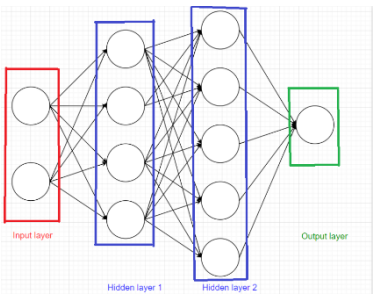
Deep regression không chỉ giỏi ở việc xấp xỉ mối quan hệ tuyến tính mà còn ở việc hiểu và mô hình hóa các mối quan hệ phi tuyến tính và phức tạp. Sự linh hoạt trong việc học các đặc trưng không chỉ làm cho mô hình có khả năng ứng phó với sự đa dạng của dữ liệu, mà còn làm nổi bật khả năng tìm kiếm các biểu diễn phức tạp và quan trọng.

- Tối ưu hóa thành công trong xử lí dữ liệu phức tạp

Deep regression vươn lên như một giải pháp xuất sắc khi đối mặt với dữ liệu phức tạp và đa dạng. Khả năng của nó trong việc học các biểu diễn tinh tế của mối quan hệ giữa các biến đầu vào và đầu ra giúp nó hiệu quả trong việc xử lý những tình huống thực tế đầy thách thức.

- Đối mặt với thách thức Overfitting

Tuy nhiên, sự linh hoạt cao của mô hình cũng đồng nghĩa với thách thức của overfitting. Quá trình điều chỉnh mô hình để giảm thiểu overfitting là một phần quan trọng của quá trình huấn luyện. Điều này đặt ra yêu cầu cao về sự kiểm soát và đánh giá cẩn thận của hiệu suất mô hình.



Hình 2. Mô hình neural network

3. TRIỂN KHAI

A. *Cấu trúc mô hình*

Mô hình mạng học sâu được xây dựng trong báo cáo sử dụng kiến trúc Sequential trong thư viện Keras để xây dựng một mạng nơ-ron tiếp tục. Cấu trúc mô hình bao gồm các lớp Dense (fully connected layers) với các hàm kích hoạt relu và một lớp cuối cùng không có hàm kích hoạt để đưa ra đầu ra.

Cấu trúc của mô hình là như sau:

Lớp Dense với 512 đơn vị nơ-ron và hàm kích hoạt relu. Lớp này có input_shape là (11,), cho biết rằng dữ liệu đầu vào có kích thước là 11 chiều.

Lớp Dense với 128 đơn vị nơ-ron và hàm kích hoạt relu.

Lớp Dense tiếp theo cũng có 128 đơn vị nơ-ron và hàm kích hoạt relu.

Lớp Dense với 32 đơn vị nơ-ron và hàm kích hoạt relu.

Lớp Dense cuối cùng với 16 đơn vị nơ-ron và không có hàm kích hoạt được áp dụng.

Lớp cuối cùng không có hàm kích hoạt, tức là đầu ra của mô hình sẽ là kết quả của lớp trước đó, không qua bất kỳ hàm biến đổi nào.

Tổng cộng, mô hình có 6 lớp, bao gồm 5 lớp Dense với hàm kích hoạt relu và một lớp cuối cùng không có hàm kích hoạt. Mô hình này có thể được sử dụng để giải quyết các bài toán phân loại với đầu ra là 2 lớp.

```
model = Sequential([
    Dense(512, activation='relu', input_shape=(11,)),
   # Dense(256, activation='relu'),
    Dense(128, activation='relu'),
    Dense(128, activation='relu'),
    Dense(32, activation='relu'),
    Dense(16, activation='relu'),
    Dense(2),
])
```

Hình 3. Code thể hiện các lớp của mô hình

Mô hình trên sử dụng mạng nơ-ron tiếp tục (feedforward neural network) là một kiểu mô hình học máy phổ biến trong đó thông tin di chuyển từ lớp đầu vào qua các lớp trung gian cho đến lớp đầu ra mà không có sự phụ thuộc ngược lại.

Cấu trúc của mạng nơ-ron tiếp tục bao gồm các lớp nơ-ron, trong đó mỗi lớp được kết nối đầy đủ với lớp tiếp theo. Mỗi nơ-ron trong lớp kế tiếp nhận đầu vào từ tất cả các nơ-ron trong lớp trước đó. Các kết nối này có trọng số, và quá trình học máy nhằm điều chỉnh các trọng số này để mô hình có thể học cách biểu diễn và rút trích thông tin từ dữ liệu đầu vào.

Mỗi nơ-ron trong mạng nơ-ron tiếp tục thực hiện hai bước chính: tổng hợp và kích hoạt. Trong bước tổng hợp, nơ-ron tính tổng có trọng số của đầu vào từ lớp trước và thêm một giá trị bias. Bước kích hoạt áp dụng một hàm kích hoạt phi tuyến tính, như hàm relu (Rectified Linear Unit), để tạo ra đầu ra của nơ-ron. Đầu ra này sau đó được chuyển tiếp đến các nơ-ron trong lớp tiếp theo và quá trình này lặp lại cho đến khi đạt được lớp đầu ra.

Mạng nơ-ron tiếp tục là một mô hình linh hoạt có thể được sử dụng cho nhiều loại bài toán, bao gồm phân loại, dự đoán và hồi quy. Điều chỉnh các kiến trúc của mạng, bao gồm số lượng lớp và số lượng nơ-ron trong mỗi lớp, có thể ảnh hưởng đến khả năng của mô hình trong việc học và biểu diễn dữ liệu.

B. *Tiến hành*

Trước khi đưa dữ liệu vào mô hình, chúng ta tiến hành các bước chuẩn bị dữ liệu để đảm bảo rằng mô hình có thể học hiệu quả từ dữ liệu và đưa ra dự đoán chính xác. Dưới đây là mô tả chi tiết về quá trình chuẩn bị dữ liệu, kết hợp với mã nguồn Python:

Đọc dữ liệu: Dữ liệu được đọc từ tệp CSV với sử dụng thư viện pandas. Điều này giúp ta quản lý và khám phá dữ liệu một cách thuận tiện.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import mean_squared_error

from tensorflow.keras.optimizers.schedules import ExponentialDecay
from sklearn.ensemble import BaggingRegressor

# Đọc dữ liệu từ CSV vào DataFrame
data = pd.read_csv('/kaggle/input/throughput-prediction/datasetNov1_data_full.csv')
```

Hình 4. Code thể hiện đọc dữ liệu

Chuẩn hóa dữ liệu: Chuẩn hóa dữ liệu là một bước quan trọng để đảm bảo rằng các biến đầu vào và đầu ra có cùng phạm vi giá trị, giúp mô hình học mối quan hệ một cách hiệu quả.

Sau khi đọc dữ liệu, chúng ta chia thành hai phần: tập dữ liệu đào tạo (X_train, y_train) và tập dữ liệu xác nhận (X_val, y_val). Điều này giúp đánh giá hiệu suất của mô hình trên dữ liệu mới mà nó chưa từng thấy.

Sau khi xây dựng xong cấu trúc mô hình, ta tiến thành biên dịch và huấn luyện mô hình:

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_30 (Dense)            (None, 512)               6144

 dense_31 (Dense)            (None, 128)               65664

 dense_32 (Dense)            (None, 128)               16512

 dense_33 (Dense)            (None, 32)                4128

 dense_34 (Dense)            (None, 16)                528

 dense_35 (Dense)            (None, 2)                 34

=================================================================
Total params: 93010 (363.32 KB)
Trainable params: 93010 (363.32 KB)
Non-trainable params: 0 (0.00 Byte)
```

Hình 5. Code thể hiện cấu trúc và thông tin Model.

Trong mô hình của nhóm tôi, có tổng cộng 93,010 tham số (hay trọng số) cần được tối ưu hóa trong quá trình huấn luyện. Tổng số tham số này đại diện cho kích thước của mô hình và cho biết mô hình có khả năng học và biểu diễn thông tin phức tạp.

Trong số các tham số đó, tất cả đều có thể được huấn luyện (trainable params). Điều này có nghĩa là trong quá trình huấn luyện, các giá trị của các tham số này sẽ được điều chỉnh để mô hình có thể tìm ra các quy tắc và mẫu trong dữ liệu huấn luyện.

Non-trainable params (tham số không thể huấn luyện) có giá trị là 0. Điều này có nghĩa là không có tham số nào trong mô hình của nhóm tôi được đặt là không thể huấn luyện và có giá trị cố định.

.

Thực hiện Train mô hình với 500 epochs và batch size là 128.

```
651/651 [==============================] - 4s 6ms/step - loss: 5.4354e-04 - val_loss: 5.5924e-04
Epoch 10/500
651/651 [==============================] - 4s 6ms/step - loss: 6.1320e-04 - val_loss: 3.4496e-04
Epoch 11/500
651/651 [==============================] - 4s 6ms/step - loss: 4.2779e-04 - val_loss: 0.0018
Epoch 12/500
651/651 [==============================] - 4s 6ms/step - loss: 4.7738e-04 - val_loss: 6.3199e-04
Epoch 13/500
...
Epoch 499/500
651/651 [==============================] - 4s 6ms/step - loss: 2.2603e-06 - val_loss: 8.8997e-06
Epoch 500/500
651/651 [==============================] - 4s 6ms/step - loss: 2.2680e-06 - val_loss: 8.7397e-06
```

Hình 6. Code thể hiện quá trình train Model.

4. KẾT QUẢ

Để đánh giá hiệu năng của mô hình, chúng ta sử dụng chỉ số RMSE (Root Mean Square Error). RMSE (Root Mean Square Error) là một chỉ số quan trọng để đánh giá hiệu năng của mô hình trong các bài toán dự đoán và hồi quy. RMSE tính toán sai số trung bình bình phương giữa các dự đoán của mô hình và giá trị thực tế.

RMSE được tính bằng cách lấy căn bậc hai của trung bình cộng của bình phương của sai số giữa dự đoán và giá trị thực tế. Giá trị RMSE càng nhỏ, tức là sai số càng thấp và mô hình có khả năng dự đoán chính xác hơn.

RMSE thường được sử dụng trong các bài toán như dự đoán giá cổ phiếu, dự đoán giá nhà, dự đoán doanh thu, và nhiều bài toán hồi quy khác

Công thức tính RMSE:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

Giải thích công thức:

- $RMSE$ là giá trị lỗi trung bình bình phương (Root Mean Square Error).
- y_i là giá trị thực tế của mẫu thứ i.
- $\hat{y}_i$ là giá trị dự đoán của mẫu thứ i.
- n là số lượng mẫu trong tập dữ liệu.

Kết quả RMSE của mô hình nhóm tôi xây dựng:

```
2602/2602 [==============================] - 4s 2ms/step
MSE: 8.739675335408276e-06
RMSE trên tập validation: 0.002956294189590792
```

Hình 7. Thể hiện thông tin kết quả RMSE

MSE (Mean Squared Error) có giá trị là 8.739675335408276e-06. MSE là một phép đo sai số bình phương trung bình giữa các dự đoán của mô hình và giá trị thực tế trên tập validation. Giá trị MSE này khá nhỏ, cho thấy mô hình có khả năng dự đoán chính xác và gần với giá trị thực tế trên tập validation.

RMSE (Root Mean Square Error) trên tập validation có giá trị là 0.002956294189590792. Giá trị RMSE này cũng khá nhỏ, cho thấy mô hình có khả năng dự đoán chính xác và gần với giá trị thực tế trên tập validation.

Cả hai giá trị MSE và RMSE đều nhỏ, điều này cho thấy mô hình đạt được kết quả tốt trên tập validation. Một RMSE thấp hơn càng tốt, vì nó chỉ ra rằng mô hình dự đoán gần với giá trị thực tế trên tập validation. Tuy nhiên, để đánh giá toàn diện hiệu suất của mô hình, cần xem xét các chỉ số và tham khảo thêm thông tin về bối cảnh và yêu cầu của bài toán cụ thể.

5. TÀI LIỆU THAM KHẢO

[1] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in CVPR, 2009. [Online].

Available: *https://ieeexplore.ieee.org/abstract/document/5206848*

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015. [Online].

Available: *https://www.nature.com/articles/nature14539*

[3] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.

[4] F. Chollet, "Deep Learning with Python," Manning Publications, 2017.