

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN CHUỖI THỜI GIAN

Dự báo hạn hán tại Charlottetown
bằng mô hình LSTM và Transformer

Nguyễn Bá Hùng - 20216835

Ngành: Toán Tin

Giảng viên hướng dẫn: TS. Nguyễn Thị Ngọc Anh

Chữ kí của GVHD

Viện:

Toán ứng dụng và Tin học

HÀ NỘI, 1/2025

Lời cảm ơn

Em xin chân thành gửi lời cảm ơn đến TS.Nguyễn Thị Ngọc Anh, người đã trang bị cho chúng em tất cả những kiến thức, kỹ năng cần có để em có thể hoàn thành tốt đề án này. Với điều kiện, thời gian cũng như kinh nghiệm còn hạn chế, đề án này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của cô để em có điều kiện bổ sung, nâng cao ý thức, phục vụ tốt hơn cho việc học tập và công tác sau này.

Em xin chân thành cảm ơn!

Tóm tắt nội dung luận văn

Đề án của em tập trung vào việc khám phá và ứng dụng ba mô hình học sâu phổ biến trong xử lý dữ liệu chuỗi thời gian, bao gồm LSTM và Transformer.

Trong **Chương 2**, Em đã trình bày cơ sở lý thuyết của từng mô hình. Chương này giải thích chi tiết về kiến trúc và nguyên lý hoạt động của Long Short-Term Memory (LSTM) và Transformer. Mỗi mô hình được phân tích về ưu và nhược điểm trong các tình huống khác nhau, đặc biệt là trong xử lý và phân tích dữ liệu chuỗi thời gian.

Chương 3 là phần thực hành, cài đặt và chạy thử nghiệm hai thuật toán trên với bộ dữ liệu thu thập từ trạm khí tượng Charlottetown.

Từ khóa: *Long Short-Term Memory (LSTM), Transformer, Time Series Forecasting.*

Hà Nội, ngày 14 tháng 1 năm 2025

Học viên thực hiện

Nguyễn Bá Hùng

MỤC LỤC

1	MỞ ĐẦU	1
1.1	Lý do chọn đề tài	1
2	CƠ SỞ LÝ THUYẾT	2
2.0.1	Mô hình LSTM	2
2.0.2	Mô hình Transformer	8
3	CÀI ĐẶT VÀ ĐÁNH GIÁ KẾT QUẢ	14
3.0.1	Tổng quan về bộ dữ liệu	14
3.0.2	Các metric đánh giá	15
3.0.3	Kết quả của mô hình LSTM	16
3.0.4	Kết quả của mô hình Transformer	19
4	KẾT LUẬN	23

DANH MỤC HÌNH VẼ

Hình 2.1	Mô hình RNN	3
Hình 2.2	Hàm Sigmoid	4
Hình 2.3	Hàm tanh	4
Hình 2.4	Kiến trúc của LSTM	5
Hình 2.5	Cổng Forget gate	6
Hình 2.6	Cổng Input gate	6
Hình 2.7	Cập nhật C_t	7
Hình 2.8	Cổng Output gate	8
Hình 2.9	Kiến trúc mô hình Transformer	9
Hình 2.10	Cơ chế Scaled Dot-Product Attention và Multi-Head Attention	10
Hình 2.11	Chi tiết một lớp Encoder	11
Hình 2.12	Lớp đầu ra của Decoder	12
Hình 3.1	Giá trị thực tế và dự báo của mô hình trong các khoảng thời gian khác nhau.	17
Hình 3.2	Đồ thị mất mát của mô hình theo các khoảng thời gian khác nhau.	18
Hình 3.3	Giá trị thực tế và dự báo của mô hình Transformer trong các khoảng thời gian khác nhau.	20
Hình 3.4	Đồ thị mất mát của mô hình theo các khoảng thời gian khác nhau	21

DANH MỤC BẢNG BIỂU

Bảng 3.1	Kết quả các chỉ số cho mô hình LSTM theo các khoảng thời gian huấn luyện và kiểm thử	19
Bảng 3.2	Kết quả các chỉ số cho mô hình Transformer theo các khoảng thời gian huấn luyện và kiểm thử	22

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt	Ý nghĩa
-------------	---------

LSTM	Long short-term memory
------	------------------------

RNN	Recurrent Neural Network
-----	--------------------------

CHƯƠNG 1. MỞ ĐẦU

1.1 Lý do chọn đề tài

Hạn hán là một thảm họa thiên nhiên khắc nghiệt, không chỉ gây tổn thất nghiêm trọng cho sản xuất nông nghiệp mà còn tạo áp lực lớn lên công tác quản lý tài nguyên nước. Tại Charlottetown, nơi phần lớn nguồn nước phụ thuộc vào lượng mưa tự nhiên, hạn hán trở thành một vấn đề cấp bách, đòi hỏi sự cải thiện trong dự báo để giảm thiểu những ảnh hưởng tiêu cực. Do đó, em đã chọn đề tài này để làm báo cáo đồ án môn học với hy vọng mang đến một góc nhìn mới trong nghiên cứu dự báo hạn hán.

Báo cáo tập trung khai thác tiềm năng của hai mô hình học sâu hiện đại – **LSTM** (Long Short-Term Memory) và **Transformer** – trong dự báo chỉ số hạn hán SPEI₁₂ (Standardized Precipitation-Evapotranspiration Index chu kỳ 12 tháng) tại Charlottetown.

Trong đó, LSTM nổi bật nhờ khả năng lưu trữ và xử lý dữ liệu chuỗi thời gian, giúp mô hình nhận diện các mẫu biến đổi dài hạn. Bên cạnh đó, Transformer, nhờ vào cơ chế *attention* mạnh mẽ, có thể nắm bắt các mối quan hệ phi tuyến và phức tạp giữa các yếu tố dữ liệu. Em kỳ vọng rằng hai mô hình này sẽ không chỉ khắc phục những hạn chế của các phương pháp truyền thống mà còn cung cấp các dự báo chính xác hơn, đặc biệt với những nhiệm vụ dự báo đa bước.

Ngoài ra, em cũng tích hợp các kỹ thuật xử lý dữ liệu chuỗi thời gian và phương pháp lựa chọn đặc trưng, nhằm tối ưu hóa hiệu quả của mô hình. Mục tiêu cuối cùng là phát triển một công cụ dự báo hạn hán có độ chính xác cao và khả năng ứng dụng thực tiễn lớn.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

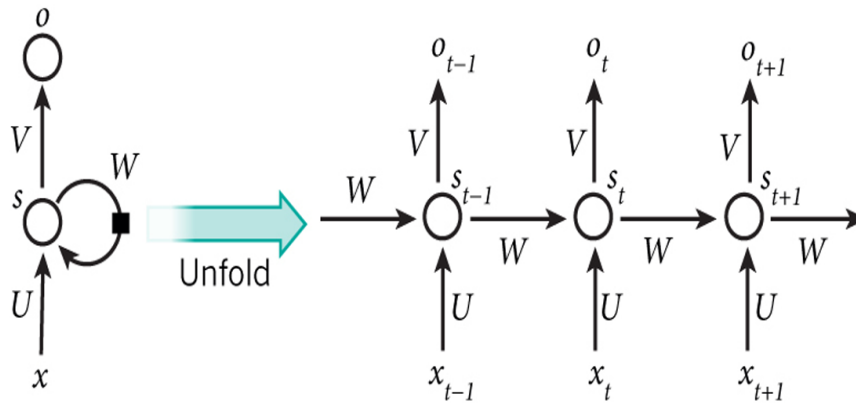
2.0.1 Mô hình LSTM

Mô hình RNN

Một mạng nơ-ron gồm những thành phần sau:

- Đầu vào- input layer
- Tầng ẩn- hidden layers
- Đầu ra- output layer,
- Các tham số trọng lượng tương ứng W , U và sai số b
- Hàm kích hoạt cho tầng ẩn

Trong mạng nơ-ron truyền thống, giả định rằng tất cả các đầu vào và đầu ra là độc lập với nhau. Điều này gây khó khăn trong các bài toán dự đoán thực tế, ví dụ như dự đoán từ tiếp theo của câu "Hãy gọi qua số điện", RNN có thể đoán ra từ tiếp theo là "thoại", điều này nơ-ron truyền thống không thể thực hiện được. Mạng RNN ra đời khắc phục nhược điểm này. Mạng RNN là có một "bộ nhớ" ghi lại thông tin về những gì đã được tính toán trong quá khứ đến nay nhưng bị hạn chế chỉ nhớ những dữ liệu gần, khả năng nắm bắt thông tin ngắn hạn.



Hình 2.1 Mô hình RNN

Giải thích sơ lược:

- x_t là input đầu vào tại t .
- s_t là trạng thái ẩn tại t , được tính toán dựa trên cả các trạng thái ẩn phía trước và đầu vào tại t :

$$s_t = f(Ux_t + Ws_{t-1}).$$

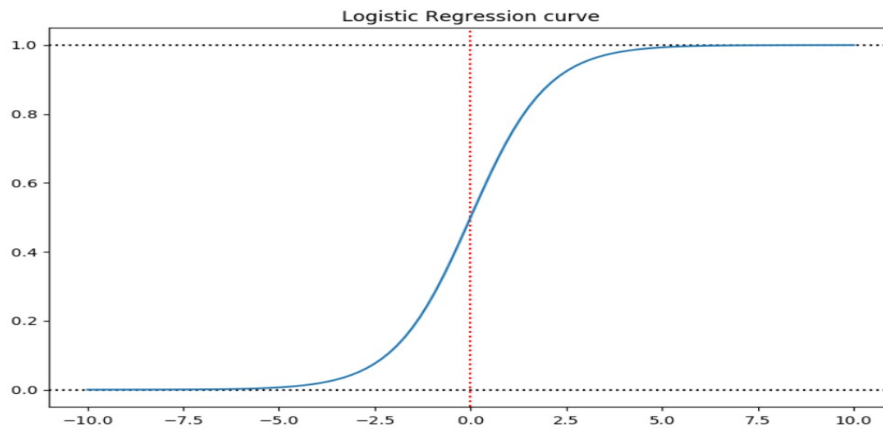
Hàm f là hàm kích hoạt.

- o_t là output đầu ra tại t .

Hàm kích hoạt

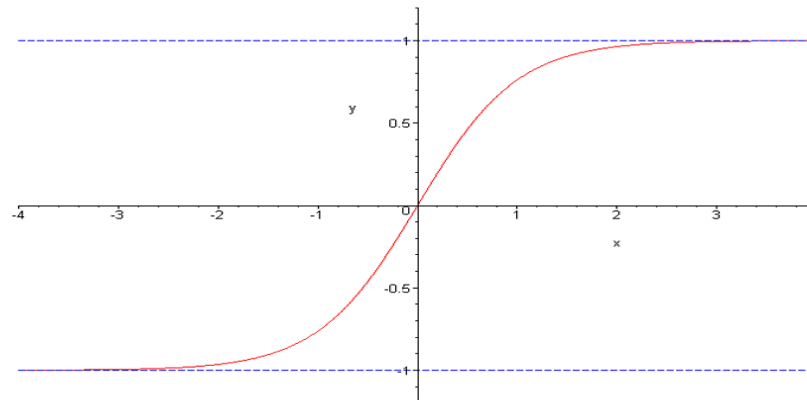
Hàm kích hoạt (activation functions) là những hàm phi tuyến được áp dụng vào đầu ra của các nơ-ron trong tầng ẩn của một mô hình mạng, và được sử dụng làm đầu vào cho tầng tiếp theo. Vì các nơ-ron không tự giới hạn được giá trị của chính nó, hàm activation sẽ cho biết bao nhiêu phần của các nơ-ron được xuất ra và quyết định xem liệu nơ-ron này có nên kích hoạt hay không.

Hàm phi tuyến Sigmoid $y(x) = \text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ cho ra kết quả được ràng buộc trong phạm vi giới hạn $(0, 1)$. Đầu ra của phi tuyến sigmoid y có thể được xem như là một xác suất để đưa ra dự đoán.



Hình 2.2 Hàm Sigmoid

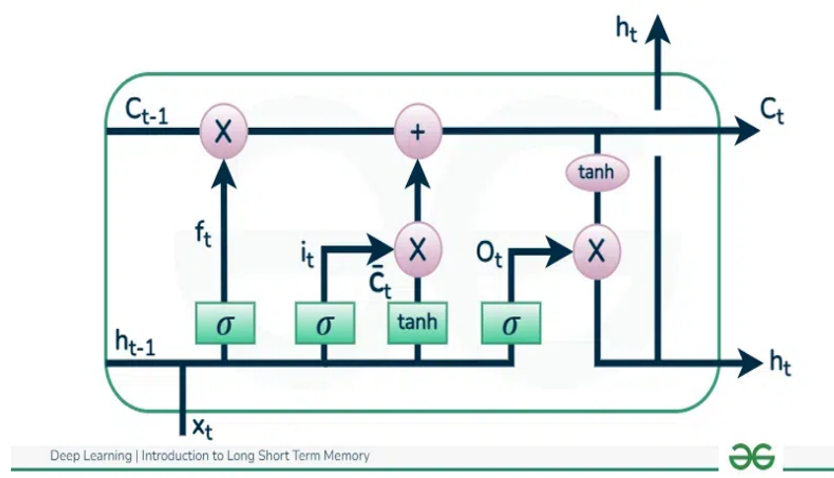
Hàm phi tuyến tanh $y(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ cho ra kết quả được ràng buộc trong phạm vi giới hạn $(-1, 1)$.



Hình 2.3 Hàm tanh

Kiến trúc của LSTM

Mô hình LSTM - một dạng của mạng Recurrent Neural Network (RNN). Chúng khắc phục nhược điểm bộ nhớ ngắn hạn của RNN, giải quyết được vấn đề vanishing-gradient và có khả năng nắm bắt các phụ thuộc dài hạn tốt hơn nhiều so với RNN thông thường. LSTM được đề xuất để xử lý nhiều trong nhiều bài toán thực tế. Thành phần LSTM gồm có một cell, với 3 cổng: một input gate i , một output gate o và một forget gate f .



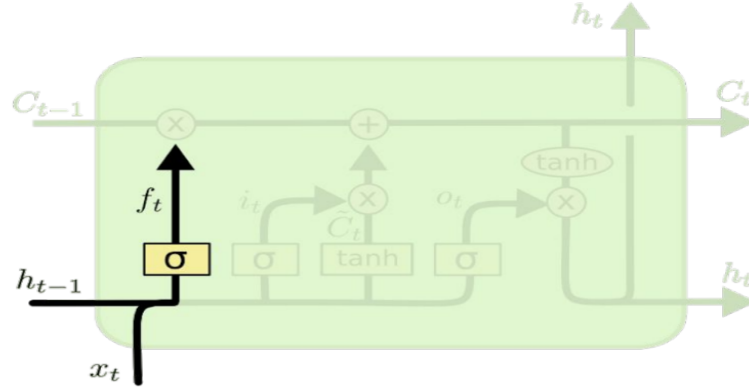
Hình 2.4 Kiến trúc của LSTM

Đầu vào là input trạng thái là x_t , trạng thái ẩn (hidden state) của trạng thái $t - 1$ là h_{t-1} và trạng thái ô (cell state) của trạng thái $t - 1$ là C_{t-1} và đầu ra là hidden state của trạng thái t là h_t và cell state của trạng thái t là C_t . Khởi đầu với $h_0 = 0$ và $C_0 = 0$, các hàm được định nghĩa như sau:

Với $W \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times m}$, $b \in \mathbb{R}^m$: các ma trận hệ số và vector sai số được học trong quá trình training. Với n là số chiều của x và m là số chiều của các vector kích hoạt. σ và \tanh lần lượt là 2 hàm kích hoạt sigmoid và tanh sẽ được dùng trong phần này.

- Cổng Forget gate sẽ quyết định chọn bao nhiêu thông tin bị bỏ quên từ trạng thái trước. Thông tin được lấy từ x_t và h_{t-1} thông qua hàm kích hoạt σ trả về giá trị $(0, 1)$.

$$f_t = \sigma(U_f \cdot x_t + W_f \cdot h_{t-1} + b_f)$$

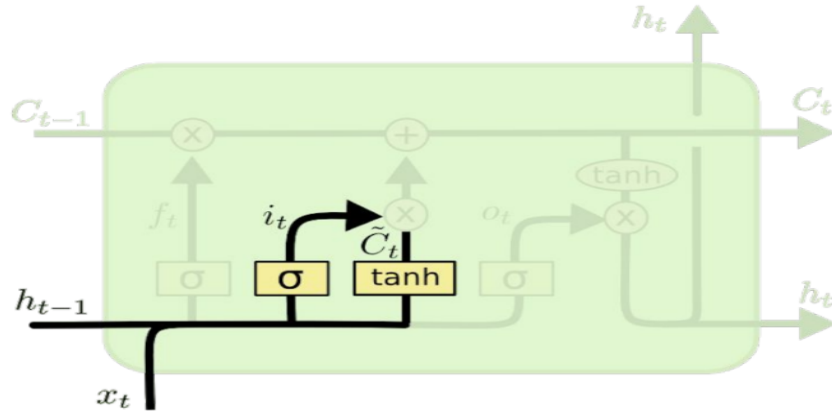


Hình 2.5 Cổng Forget gate

- Tiếp theo quyết định thông tin mới được lưu lại ở đầu vào trạng thái hiện tại. Cổng Input gate sẽ đưa thông tin state và hidden state của cổng trước qua 2 hàm: sử dụng hàm kích hoạt σ trả về giá trị $(0, 1)$ để quyết định giá trị nào sẽ cập nhật và hàm kích hoạt tanh tạo vector \tilde{C}_t mới có giá trị $(-1, 1)$ để thêm vào trạng thái.

$$i_t = \sigma(U_i \cdot x_t + W_i \cdot h_{t-1} + b_i)$$

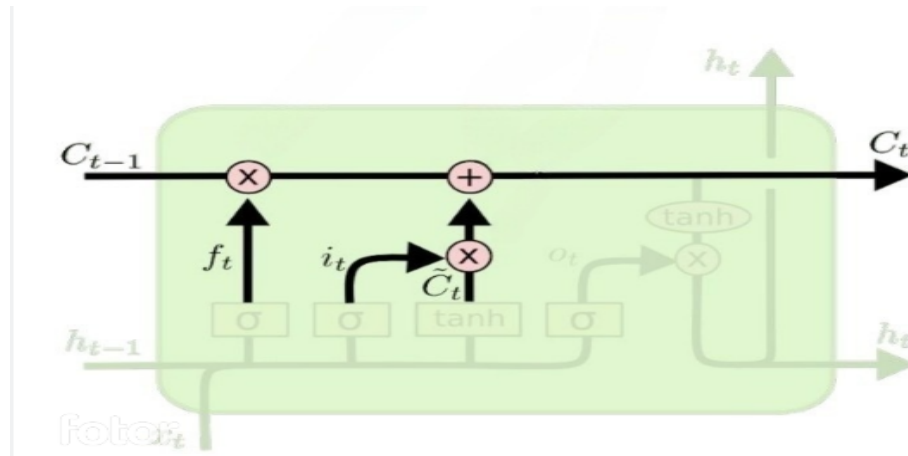
$$\tilde{C}_t = \tanh(U_c \cdot x_t + W_c \cdot h_t + b_c)$$



Hình 2.6 Cổng Input gate

- Tiếp theo là cập nhật cell state mới C_t từ các thông tin bị bỏ quên ở cell state trước $f_t \cdot C_{t-1}$ và thông tin mới được cập nhật ở đầu vào $i_t \cdot \tilde{C}_t$.

$$C_t = \sigma(f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t)$$

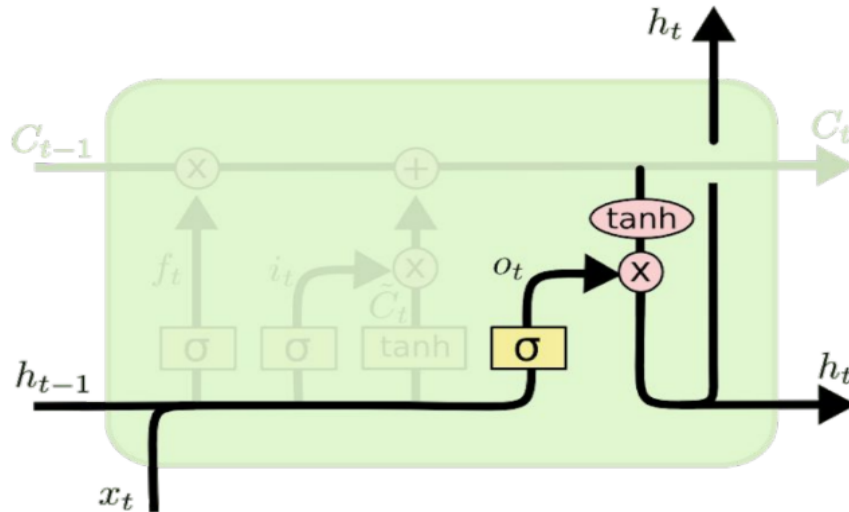


Hình 2.7 Cập nhật C_t

- Cổng Output quyết định thông tin đầu ra của trạng thái hiện tại. Cổng Output gate sẽ sử dụng hàm kích hoạt σ trả về giá trị $(0, 1)$ để quyết định chọn bao nhiêu thông tin trạng thái sẽ được xuất ra. Trạng thái mới C_t qua hàm \tanh để trả về giá trị $(-1, 1)$. Kết hợp 2 giá trị để trở thành output cho trạng thái ẩn. Kết quả đầu ra đó (hidden state) và cả cell state sẽ được đưa vào bước tiếp theo.

$$o_t = \sigma(U_o \cdot x_t + W_o \cdot h_{t-1} + b_o)$$

$$h_t = \tanh(C_t) \cdot o_t$$



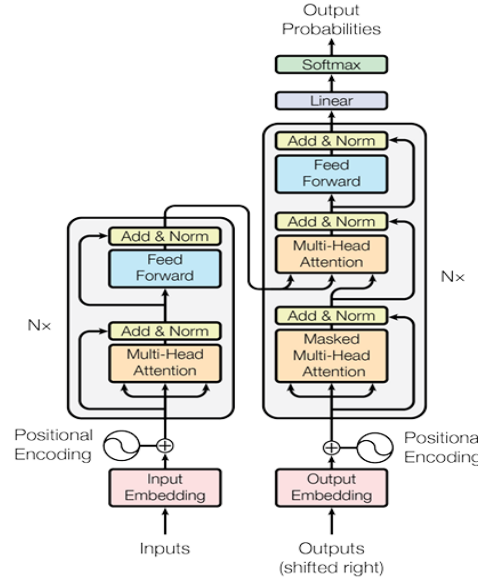
Hình 2.8 Cổng Output gate

2.0.2 Mô hình Transformer

Transformer là một kiến trúc mạng nơ-ron được giới thiệu trong bài báo nổi tiếng “Attention is All You Need” (Vaswani et al., 2017). Nó được thiết kế để giải quyết các bài toán sequence-to-sequence như dịch máy, tóm tắt văn bản, v.v. Điểm khác biệt lớn của Transformer so với các mô hình trước đó như RNN hay CNN là nó dựa hoàn toàn vào cơ chế attention mà không sử dụng recurrence hay convolution.

-Ưu điểm chính của Transformer:

- **Khả năng song song hóa:** Tính toán song song toàn bộ chuỗi thay vì xử lý từng bước như RNN, giúp tăng tốc độ huấn luyện.
- **Xử lý mối quan hệ dài hạn:** Cơ chế attention giúp mô hình nắm bắt tốt các mối quan hệ xa giữa các phần tử trong chuỗi dữ liệu.
- **Tính tổng quát cao:** Transformer phù hợp với nhiều tác vụ không chỉ trong xử lý ngôn ngữ tự nhiên mà còn các lĩnh vực như thị giác máy tính và dự báo chuỗi thời gian.



Hình 2.9 Kiến trúc mô hình Transformer

1. Input Embedding và Positional Encoding

Mỗi token trong chuỗi đầu vào được chuyển đổi thành một vector bằng lớp nhúng (embedding). Để mô hình nắm được thứ tự các token trong chuỗi, **Positional Encoding** được thêm vào biểu diễn nhúng này.

-Công thức Positional Encoding:

$$PE_{\text{pos},2i} = \sin\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \quad PE_{\text{pos},2i+1} = \cos\left(\frac{\text{pos}}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$

Trong đó:

- pos: Vị trí token trong chuỗi.
- i : Thứ tự chiều trong vector nhúng.
- d_{model} : Kích thước vector nhúng.

Vector đầu vào sau khi cộng Positional Encoding:

$$x_{\text{input}} = x_{\text{embedding}} + x_{\text{positional encoding}}$$

2. Encoder

2.1 Multi-Head Self-Attention:

Multi-Head Self-Attention là cơ chế quan trọng giúp mô hình học được mối quan hệ giữa các token trong chuỗi đầu vào.

Scaled Dot-Product Attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

Trong đó:

- Q : Ma trận queries.
- K : Ma trận keys.
- V : Ma trận values.
- d_k : Kích thước vector key.

Multi-Head Attention:

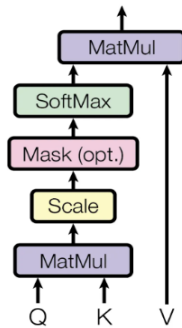
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

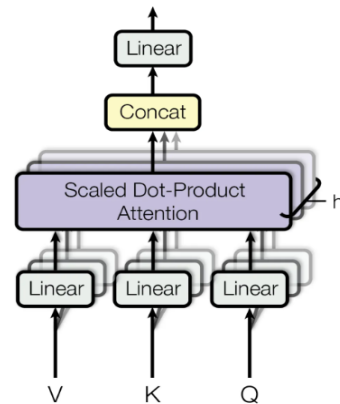
Trong đó:

- h : Số lượng đầu attention.
- W_i^Q, W_i^K, W_i^V : Các ma trận ánh xạ cho từng đầu.
- W^O : Ma trận ánh xạ cuối cùng để ghép nối đầu ra các head.

Scaled Dot-Product Attention



Multi-Head Attention

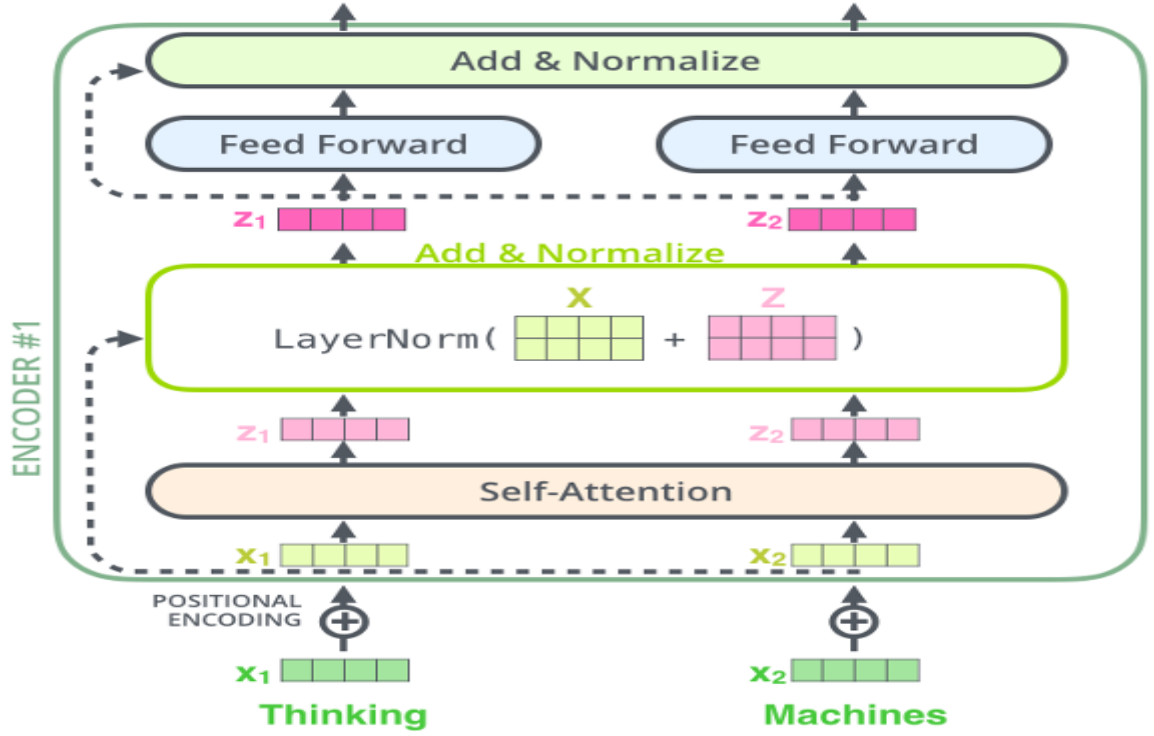


Hình 2.10 Cơ chế Scaled Dot-Product Attention và Multi-Head Attention

2.2 Residual Connection và Layer Normalization

Mỗi thành phần chính trong encoder đều được bọc bởi residual connection để tránh mất mát thông tin và được chuẩn hóa bằng layer normalization:

$$\text{Output}_{\text{attention}} = \text{LayerNorm}(\text{Input} + \text{MultiHeadAttention})$$



Hình 2.11 Chi tiết một lớp Encoder

2.3 Position-wise Feed-Forward Network (FFN)

Position-wise FFN được áp dụng độc lập cho từng vị trí trong chuỗi:

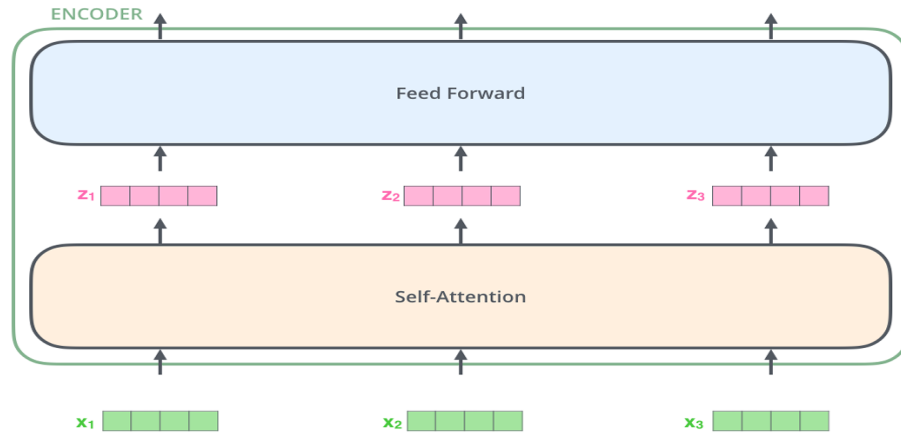
$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2$$

Trong đó:

- W_1, W_2 : Các ma trận trọng số.
- b_1, b_2 : Các vector bias.
- ReLU: Hàm kích hoạt.

Đầu ra được kết hợp với đầu vào qua residual connection và chuẩn hóa:

$$\text{Output}_{\text{encoder}} = \text{LayerNorm}(\text{Attention Output} + \text{FFN Output})$$



3. Decoder

Decoder bổ sung một thành phần quan trọng là **Masked Multi-Head Self-Attention**, để đảm bảo mô hình chỉ sử dụng thông tin từ các token trước đó và hiện tại khi tạo ra đầu ra.

3.1 Multi-Head Encoder-Decoder Attention

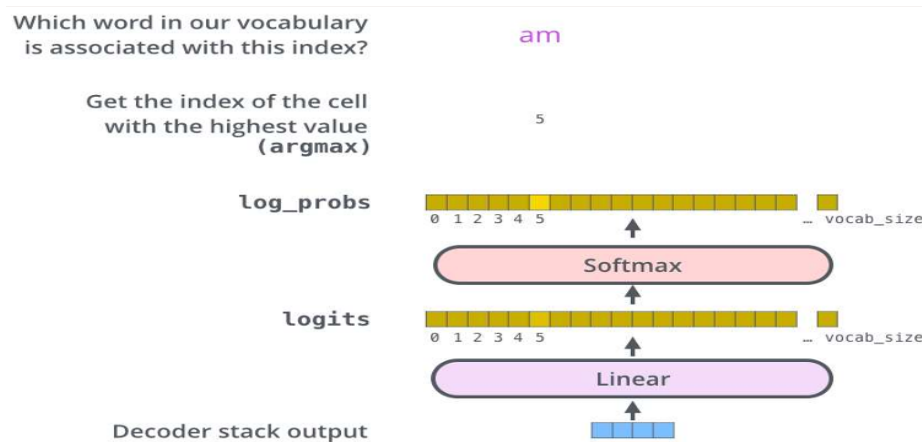
Cơ chế này cho phép mỗi token trong decoder tập trung vào toàn bộ đầu ra của encoder:

$$\text{Attention}_{\text{encoder-decoder}} = \text{softmax} \left(\frac{Q_{\text{decoder}} K_{\text{encoder}}^T}{\sqrt{d_k}} \right) V_{\text{encoder}}$$

4. Output Layer

Đầu ra từ decoder được chuyển đổi qua lớp tuyến tính và hàm softmax để dự đoán xác suất của từ tiếp theo trong chuỗi:

$$\text{Probability}(y_t) = \text{softmax}(zW^T + b)$$



Hình 2.12 Lớp đầu ra của Decoder

5. Learning Rate Scheduling

Tốc độ học được điều chỉnh theo bước huấn luyện để ổn định quá trình học:

$$\text{lrate} = d_{\text{model}}^{-0.5} \cdot \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup_steps}^{-1.5})$$

Trong đó:

- `step`: Bước huấn luyện hiện tại.
- `warmup_steps`: Số bước ban đầu để tăng tốc độ học.
- `dmodel`: Số chiều của vector nhúng.

CHƯƠNG 3. CÀI ĐẶT VÀ ĐÁNH GIÁ KẾT QUẢ

3.0.1 Tổng quan về bộ dữ liệu

Bộ dữ liệu ban đầu được thu thập từ hai trạm khí tượng Charlottetown bao gồm chỉ số hạn hán SPEI₁₂ trong giai đoạn 1980-2022. Bộ dữ liệu được chia thành các file sau:

- chars1.csv: Chứa dữ liệu tín hiệu lịch sử và mục tiêu dự đoán chỉ số SPEI₁₂ cho 1 tháng tiếp theo.
- chars3.csv: Chứa dữ liệu tín hiệu lịch sử và mục tiêu dự đoán chỉ số SPEI₁₂ cho 3 tháng tiếp theo.
- chars6.csv: Chứa dữ liệu tín hiệu lịch sử và mục tiêu dự đoán chỉ số SPEI₁₂ cho 6 tháng tiếp theo.
- chars12.csv: Chứa dữ liệu tín hiệu lịch sử và mục tiêu dự đoán chỉ số SPEI₁₂ cho 12 tháng tiếp theo.

Mỗi file bao gồm các biến sau:

- T-1: Giá trị chỉ số SPEI tại thời điểm 1 tháng trước so với thời điểm cần dự đoán.
- T-2: Giá trị chỉ số SPEI tại thời điểm 2 tháng trước so với thời điểm cần dự đoán.
- T-3: Giá trị chỉ số SPEI tại thời điểm 3 tháng trước so với thời điểm cần dự đoán.
- Tar: Giá trị mục tiêu SPEI₁₂ tại thời điểm cần dự đoán.

Bằng việc phân tích EMD (Empirical Mode Decomposition) ta sẽ có được các file chứa thông tin chi tiết hơn về các thành phần tần số trong chuỗi thời gian là Chartvf1.csv, Chartvf3.csv, Chartvf6.csv, Chartvf12.csv.

Các biến IMF1 đến IMF11: Mỗi biến IMF tương ứng với một tần số riêng biệt trong tín hiệu chuỗi thời gian. Các giá trị này được ghi nhận theo thời gian trễ từ IMF1($t-1$), IMF1($t-2$), IMF1($t-3$), cho đến IMF11($t-1$), IMF11($t-2$), IMF11($t-3$), giúp làm rõ hơn các biến đổi tần số của dữ liệu qua từng giai đoạn.

3.0.2 Các metric đánh giá

Mean absolute error (MAE)

Phương pháp đánh giá này sử dụng độ lệch tuyệt đối trung bình giữa các giá trị quan sát và giá trị dự đoán để đưa ra kết luận. Mọi biến động trong dữ liệu đều được coi là quan trọng như nhau trong phương pháp này. Công thức MAE được tính như sau:

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|$$

Trong đó, \hat{y}_i là giá trị dự đoán tại thời điểm i và y_i là giá trị thực tế.

Mean square error (MSE)

Sự khác biệt giữa giá trị thực tế và giá trị dự kiến của mô hình được sử dụng trong việc tính toán giá trị này, và sau đó được bình phương. Các sai lệch càng lớn, chỉ số này càng phạt nặng hơn. Công thức MSE được tính như sau:

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

Root Mean Square Error(RMSE)

RMSE đo lường trung bình độ lệch bình phương giữa giá trị thực tế và giá trị dự báo, sau đó lấy căn bậc hai. RMSE càng nhỏ, mô hình càng chính xác. Công thức RMSE được tính như sau:

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

Trong đó:

- n : Số lượng điểm dữ liệu trong tập kiểm tra.
- y_i : Giá trị thực tế tại thời điểm i .
- \hat{y}_i : Giá trị dự báo tại thời điểm i .

R-squared (R²)

là thước đo thể hiện mức độ biến động của biến mục tiêu được giải thích bởi mô hình. Công thức R^2 được tính như sau:

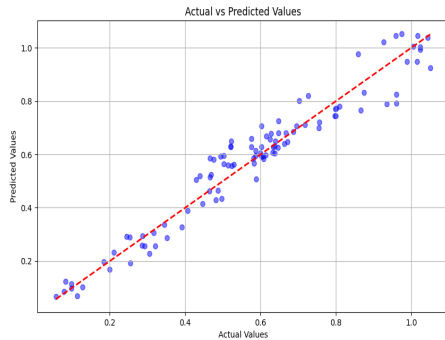
$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Trong đó:

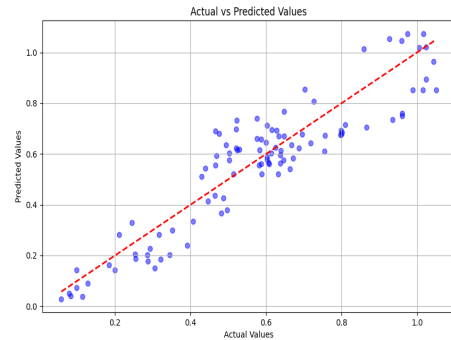
- y_i : Giá trị thực tế của biến mục tiêu tại thời điểm i .
- \hat{y}_i : Giá trị dự báo tại thời điểm i .
- \bar{y} : Giá trị trung bình của y trong tập dữ liệu.
- n : Số lượng quan sát.

3.0.3 Kết quả của mô hình LSTM

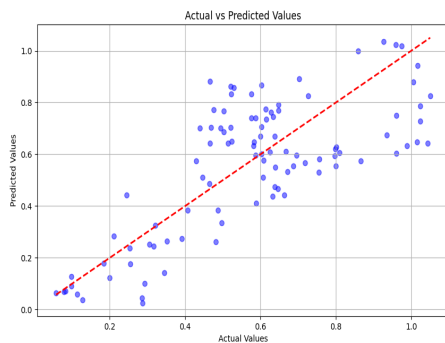
- Giá trị thực tế và giá trị dự báo của mô hình.



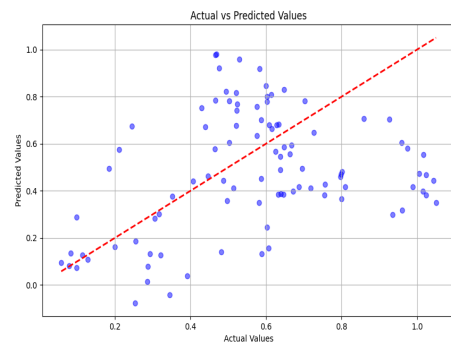
(a) Thực tế và dự báo (1 tháng).



(b) Thực tế và dự báo (3 tháng).



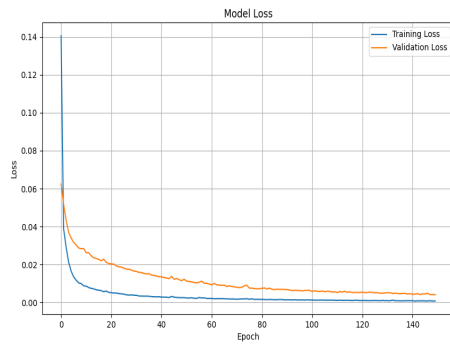
(c) Thực tế và dự báo (6 tháng).



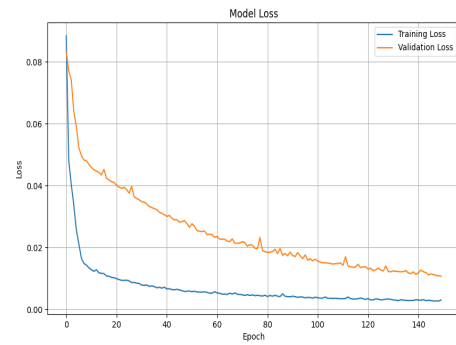
(d) Thực tế và dự báo (12 tháng).

Hình 3.1 Giá trị thực tế và dự báo của mô hình trong các khoảng thời gian khác nhau.

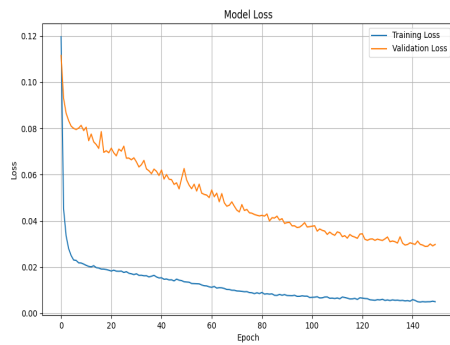
- Đồ thị mất mát của mô hình:



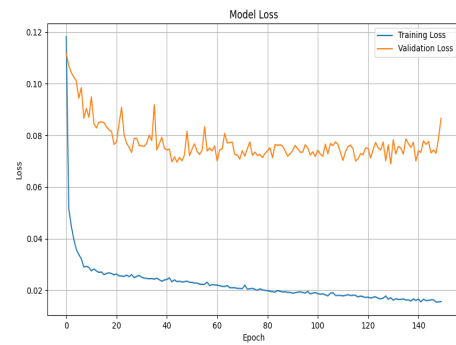
(a) 1 tháng



(b) 3 tháng



(c) 6 tháng



(d) 12 tháng

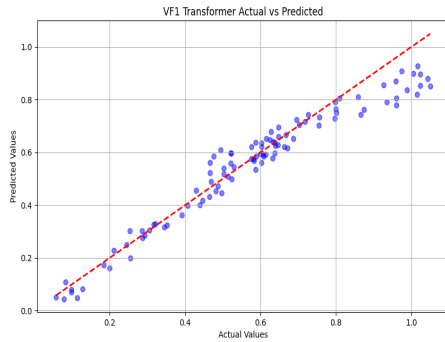
Hình 3.2 Đồ thị mất mát của mô hình theo các khoảng thời gian khác nhau.

Bảng 3.1 Kết quả các chỉ số cho mô hình LSTM theo các khoảng thời gian huấn luyện và kiểm thử .

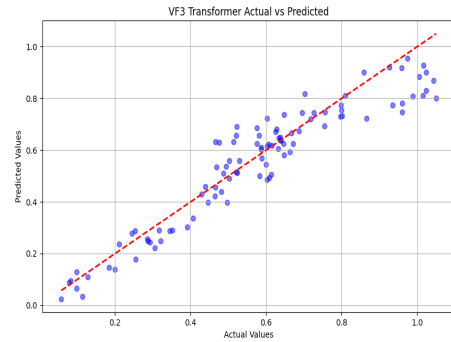
Khoảng thời gian	Tập dữ liệu	MSE	RMSE	MAE	R^2
1 tháng	Train	0.0011	0.0325	0.0252	0.9815
	Test	0.0037	0.0609	0.0477	0.9415
3 tháng	Train	0.0034	0.0582	0.0451	0.9414
	Test	0.0106	0.1028	0.0874	0.8325
6 tháng	Train	0.0075	0.0868	0.0637	0.8705
	Test	0.0332	0.1821	0.1521	0.4750
12 tháng	Train	0.0238	0.1541	0.1167	0.5834
	Test	0.0946	0.3075	0.2513	-0.5252

3.0.4 Kết quả của mô hình Transformer

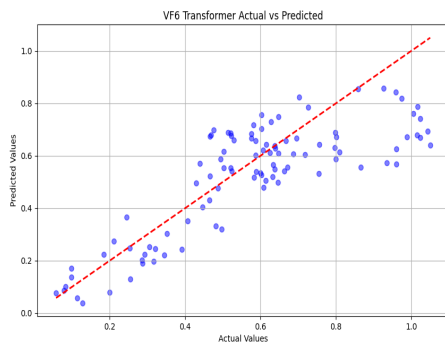
- Giá trị thực tế và giá trị dự báo của mô hình Transformer:



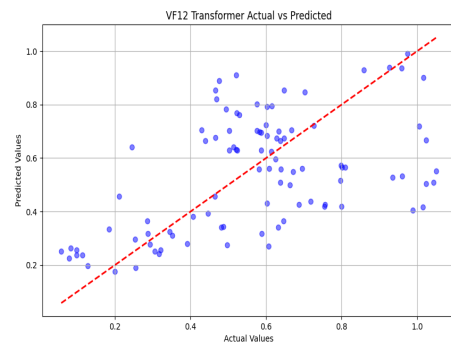
(a) Thực tế và dự báo (1 tháng).



(b) Thực tế và dự báo (3 tháng).



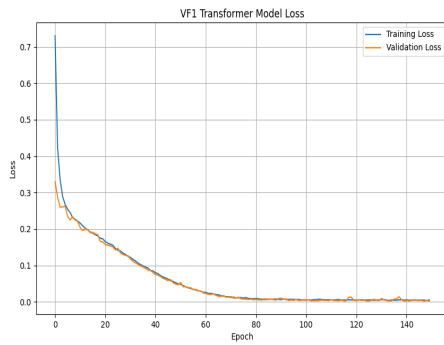
(c) Thực tế và dự báo (6 tháng).



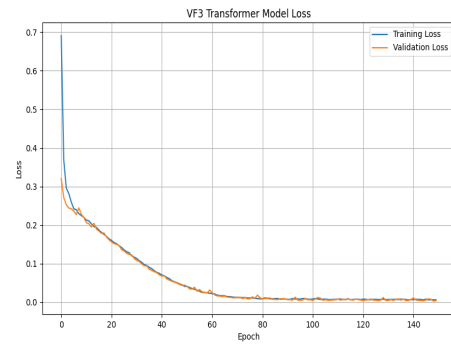
(d) Thực tế và dự báo (12 tháng).

Hình 3.3 Giá trị thực tế và dự báo của mô hình Transformer trong các khoảng thời gian khác nhau.

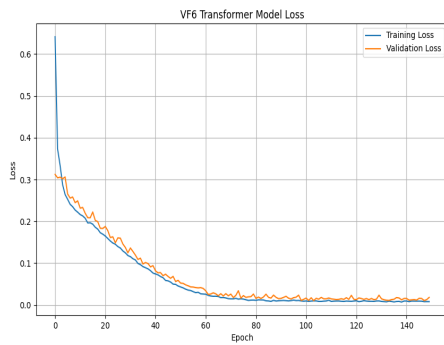
- Đồ thị mất mát của mô hình Transformer:



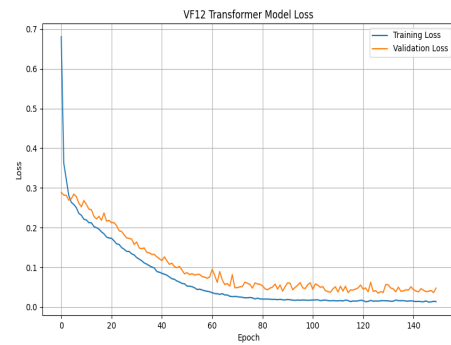
(a) 1 tháng



(b) 3 tháng



(c) 6 tháng



(d) 12 tháng

Hình 3.4 Đồ thị mất mát của mô hình theo các khoảng thời gian khác nhau .

Bảng 3.2 Kết quả các chỉ số cho mô hình Transformer theo các khoảng thời gian huấn luyện và kiểm thử .

Khoảng thời gian	Tập dữ liệu	MSE	RMSE	MAE	R^2
1 tháng	Train	0.0015	0.0389	0.0303	0.9736
	Test	0.0044	0.0663	0.0468	0.9308
3 tháng	Train	0.0020	0.0443	0.0352	0.9660
	Test	0.0072	0.0849	0.0636	0.8858
6 tháng	Train	0.0066	0.0813	0.0618	0.8864
	Test	0.0234	0.1530	0.1197	0.6294
12 tháng	Train	0.0127	0.1126	0.0849	0.7778
	Test	0.0532	0.2306	0.1813	0.1425

Nhận xét :

- Mô hình hoạt động tốt với dự báo ngắn hạn: Cả hai mô hình trên hoạt động rất tốt trong việc dự báo hạn hán cho 1 tháng. Độ chính xác tốt (MSE, RMSE, MAE nhỏ) và khả năng mô hình hóa mối quan hệ tốt (R^2 cao).
- Hiệu suất giảm dần theo thời gian: Khi khoảng thời gian dự báo tăng lên (3, 6, 12 tháng), hiệu suất của mô hình giảm rõ rệt. Điều này có thể do tính không ổn định của dữ liệu hạn hán theo thời gian, cũng như sự tích lũy của sai số khi mô hình dự báo cho các bước thời gian xa hơn.

CHƯƠNG 4. KẾT LUẬN

- Những nội dung đã đạt được:

- Giới thiệu mô hình LSTM và Transformer trong bài toán dự báo chuỗi thời gian
- Ứng dụng LSTM và Transformer vào dự đoán mức độ hạn hán

- Hướng phát triển : Hiện tại, đề án mới chỉ triển khai những thuật toán cơ bản nhất của LSTM và Transformer. Vì vậy, có một số hướng phát triển có thể thực hiện cho đề án này:

- Thay đổi các siêu tham số của các mô hình để xem xét sự phụ thuộc.
- Kết hợp các mô hình: Nghiên cứu các mô hình lai kết hợp ưu điểm của cả LSTM và Transformer để nâng cao độ chính xác dự báo.
- Sử dụng cơ chế attention: Thử nghiệm cơ chế attention vào LSTM để cải thiện khả năng dự báo dài hạn.
- Kết hợp phương pháp vật lý: Kết hợp các mô hình học máy với các phương pháp vật lý để có thể kết hợp thêm các kiến thức chuyên môn vào mô hình.

Tài liệu tham khảo

- [1] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention is all you need*. Advances in Neural Information Processing Systems, 30, 2017.
- [2] Serkan Kiranyaz, Onur Avcı, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. *1D convolutional neural networks and applications: A survey*, 2019.
- [3] Breno Aguiar Krohling and Renato A Krohling. *1D convolutional neural networks and machine learning algorithms for spectral data classification with a case study for COVID-19*, 2023.
- [4] Jun Luo and Yaping Gong. *Air pollutant prediction based on ARIMA-WOA-LSTM model*. Atmospheric Pollution Research, 14:101761, 04 2023.
- [5] Diaa Salman, Cem Direkoglu, Mehmet Kusaf, and Murat Fahrioglu. *Hybrid deep learning models for time series forecasting of solar power*. Neural Computing and Applications, 36, 02 2024.
- [6] Robin M. Schmidt. *Recurrent neural networks (RNNs): A gentle introduction and overview*, 2019.
- [7] Aunkit Chaki, Ankit Kudeshia, and Sourodip Ghosh. *Cyberbully detection using 1D-CNN and LSTM*. 728, 4 2021.
- [8] Ralf C. Staudemeyer and Eric Rothstein Morris. *Understanding LSTM– A tutorial into long short-term memory recurrent neural networks*, 2019.
- [9] Christian Bakke Vennerød, Adrian Kjærran, and Erling Stray Bugge. *Long short-term memory RNN*, 2021.