HUS
VNU UNIVERSITY OF SCIENCE

Nhận diện tin giả

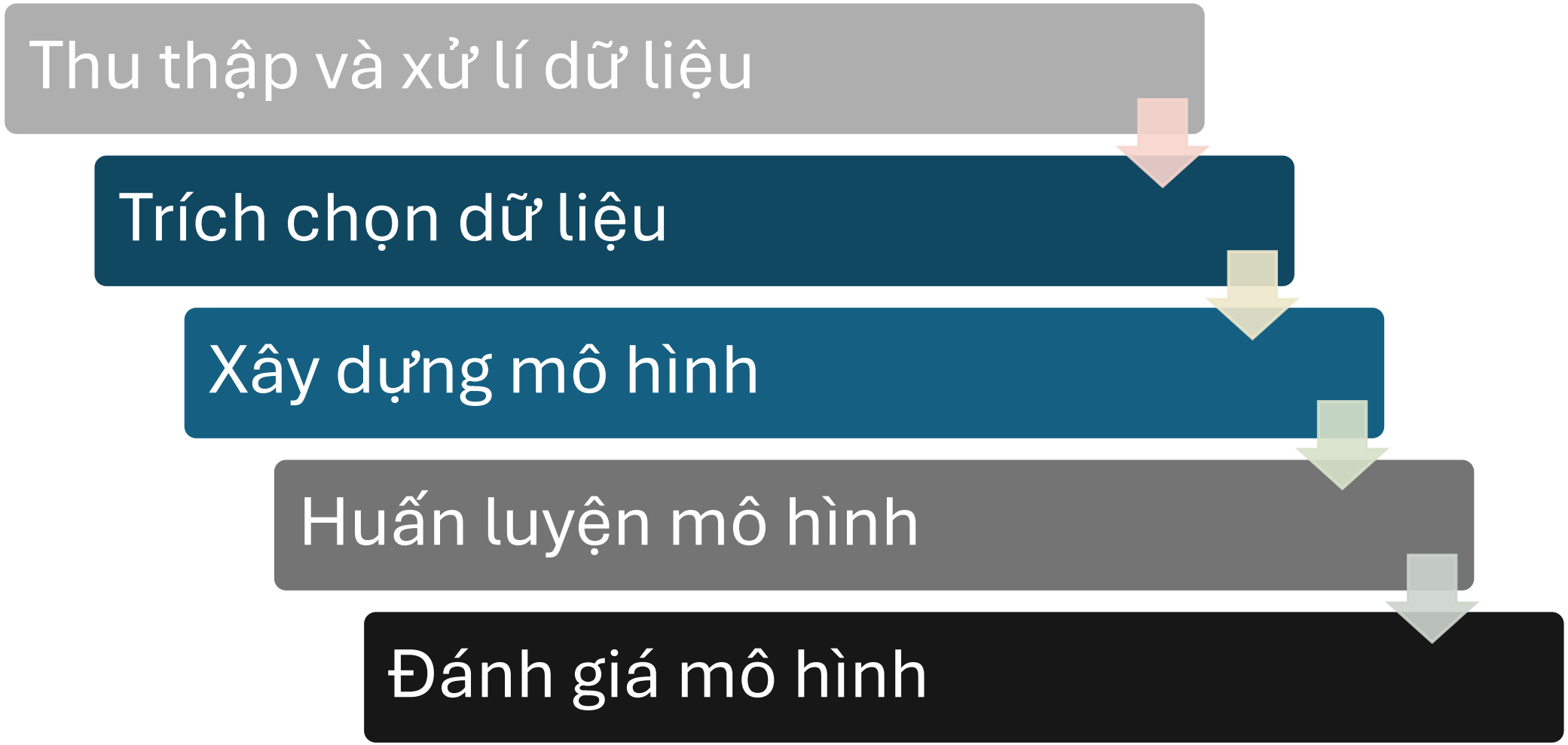
Thành viên: Nguyễn Thị Phương Hoa_SIC2253
Nguyễn Huyền Trang_SIC2281
Dương Hoàng Long_SIC2269
Bùi Đức Phú Anh_ SIC2271
Nguyễn Mai Hương_SIC2376

Mục lục

Giới thiệu

Hình 1. Tin giả

Giới thiệu


Thu thập và xử lí dữ liệu
Trích chọn dữ liệu
Xây dựng mô hình
Huấn luyện mô hình
Đánh giá mô hình

Xử lí và phân tích dữ liệu

- Dữ liệu: [Liar Preprocessed Dataset (kaggle.com)]

- Dataset của LIAR được sử dụng, đã được kiểm định và đã được tiền xử lý. File train gồm có 15 cột bao gồm: ID của bài báo, nội dung bài báo, chủ đề, người viết, chức danh người viết, bốicảnh (nơi diễn ra/địa điểm phát biểu), phần lý giải trích dẫn,...

false	Says the Annies ...	abortion	dwayne-bohac	State representa...
[null] 21% federal-budget 1% Other (8018) 78%	[null] 48% economy 1% Other (5254) 51%	[null] 70% state-budget 0% Other (2980) 29%	[null] 85% state-budget 0% Other (1520) 15%	[null] 93% history 0% Other (732) 7%
half-true	When did the decline of coal start? It started when natural gas took off that started to begin in (P...	energy,history,job-accomplishments	scott-surovell	State delegate
mostly-true	Hillary Clinton agrees with John McCain "by voting to give George Bush the benefit of the doubt on I...	foreign-policy	barack-obama	President
false	Health care reform legislation is likely to mandate free sex change surgeries.	health-care	blog-posting	
half-true	The economic turnaround started at the end of my term.	economy,jobs	charlie-crist	
true	The Chicago Bears	education	robin-vos	Wisconsin Assembly

Hình 2. Bảng dữ liệu

Xử lí và phân tích dữ liệu

- Dữ liệu của LIAR được chia làm 6 nhãn khác nhau dựa theo tính chính xác của tin tức, bao gồm:

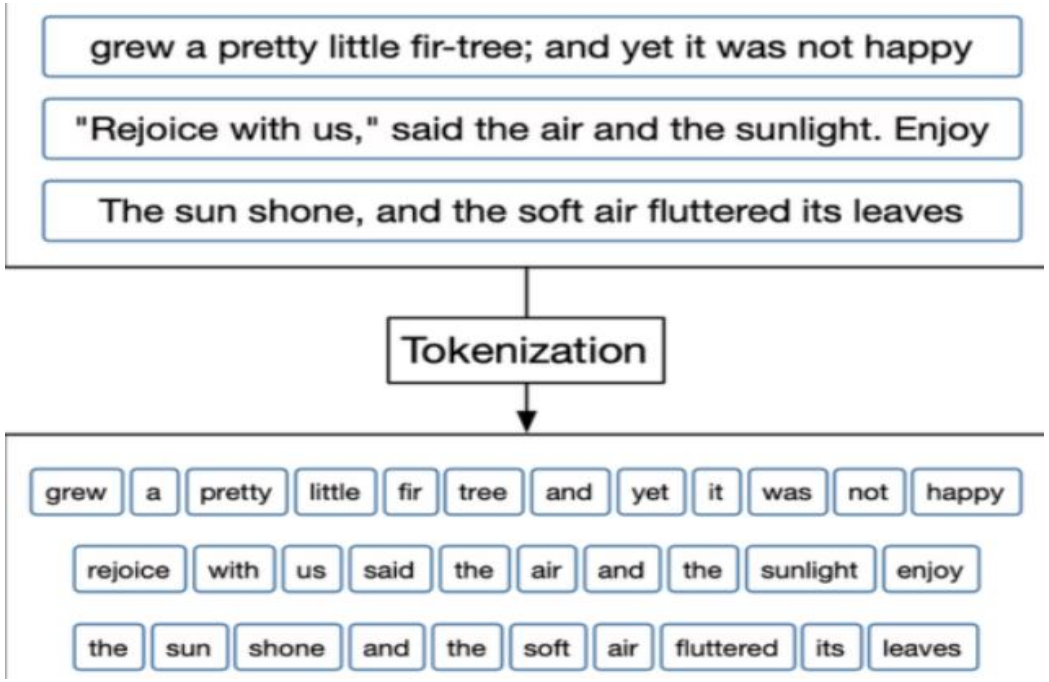
- TRUE (tin thật)
- MOSTLY-TRUE (tin gần thật)
- HALFLY-TRUE (tin nửa thật)
- BARELY TRUE (tin gần không thật)
- FALSE (tin giả)
- PANTS-FIRE (tin lừa đảo)

data	true	mostly-true	half-true	barely-true	false	pants-fire	total
train	1676	1962	2114	1654	1995	839	10240
test	208	241	265	212	249	92	1267
valid	169	251	248	237	263	116	1284

Hình 3. Phân phối nhãn trong bộ dữ liệu LIAR

Xử lí và phân tích dữ liệu

- Trước khi được đưa vào mô hình học máy, nội dung của tin mẫu được chia thành các từ nhỏ (Tokenization) và sau đó đưa về dạng cơ bản nhất (Lemmatization).
- Trong bước này ta cũng có thể phân tích được số từ, câu và kí tự trong tập dữ liệu, cụ thể trong bộ dữ liệu LIAR tính trên 12791 mẫu, trung bình mỗi mẫu có: 107.161 kí tự, 20.210 từ và 1.167 câu



Hình 4. Tokenization

	count	mean	std	min	25%	50%	75%	max
characters	12791.0	107.161520	63.452113	11.0	73.0	99.0	133.0	3192.0
words	12791.0	20.210695	11.457018	2.0	14.0	19.0	25.0	546.0
sentences	12791.0	1.167383	0.563874	1.0	1.0	1.0	1.0	19.0

Hình 5. Phân phối kí tự, câu và từ trong bộ dữ liệu LIAR

Xử lí và phân tích dữ liệu

Trích xuất đặc trưng: quá trình phân tích và lấy ra những thông tin **quan trọng** hoặc **có giá trị**

$$\boldsymbol{TF - IDF = tf * idf}$$

$$\boldsymbol{tf(t,d)} = \frac{\text{Tần suất của từ } t \text{ trong văn bản } d}{\text{Tần suất của từ xuất hiện nhiều nhất trong văn bản } d}$$

$$\boldsymbol{idf(t,D)} = \log \frac{\text{Tổng số văn bản}}{\text{Số văn bản chứa từ đang xét}}$$

Xử lí và phân tích dữ liệu

Trích xuất đặc trưng: quá trình phân tích và lấy ra những thông tin **quan trọng** hoặc **có giá trị**

Count vectorization

Là một kỹ thuật trong xử lý ngôn ngữ tự nhiên (NLP) dùng để chuyển đổi văn bản thành dạng số mà máy tính có thể hiểu được.

- Xây dựng từ điển: Mỗi từ trong tập dữ liệu văn bản được gán một chỉ số trong từ điển
- Vecto hóa: Mỗi văn bản được biểu diễn như một vecto, trong đó mỗi phần tử của vecto là số lần xuất hiện của từ trong từ điển văn bản đó

Xử lí và phân tích dữ liệu

Trích xuất đặc trưng: quá trình phân tích và lấy ra những thông tin **quan trọng** hoặc **có giá trị**

Count vectorization

Xét 3 tài liệu:
Doc 1: “I love rock music
Doc 2: “Rock music is great
Doc 3: “I love great music

Ma trận thu được

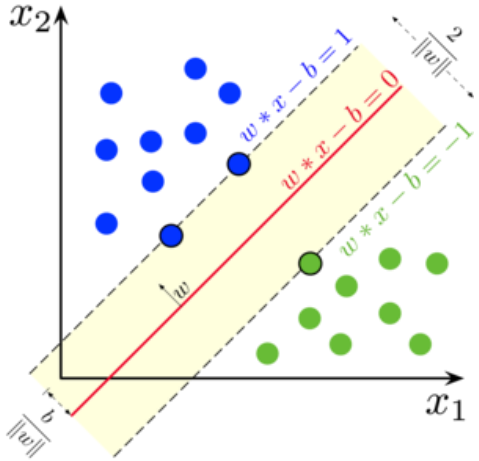
	"I"	"love"	"rock"	"music"	"is"	"great"
Doc 1	1	1	1	1	0	0
Doc 2	0	0	1	1	1	1
Doc 3	1	1	0	1	0	1

Sử dụng Count vectorization
“I”: 0 “love”: 1 “rock”: 2
“music”: 3 “is”: 4 “great”: 5

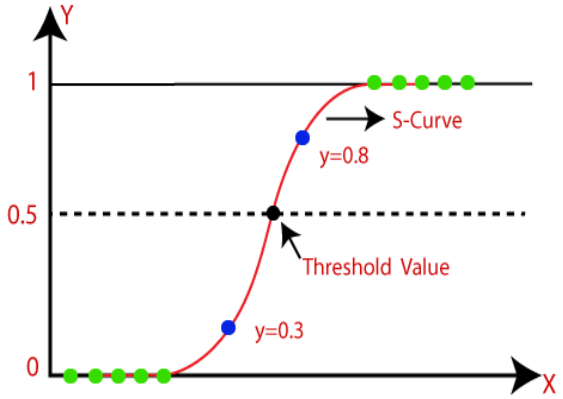
Xây dựng mô hình

Học máy

SVM

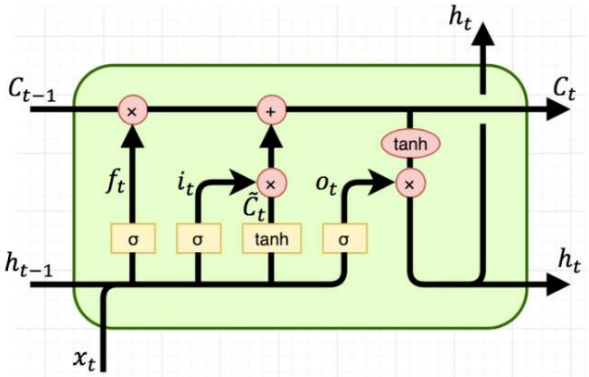


Hồi quy Logistic



Học sâu

LSTM



Transformers

Xây dựng mô hình (SVM)

1.1. Mô hình SVM

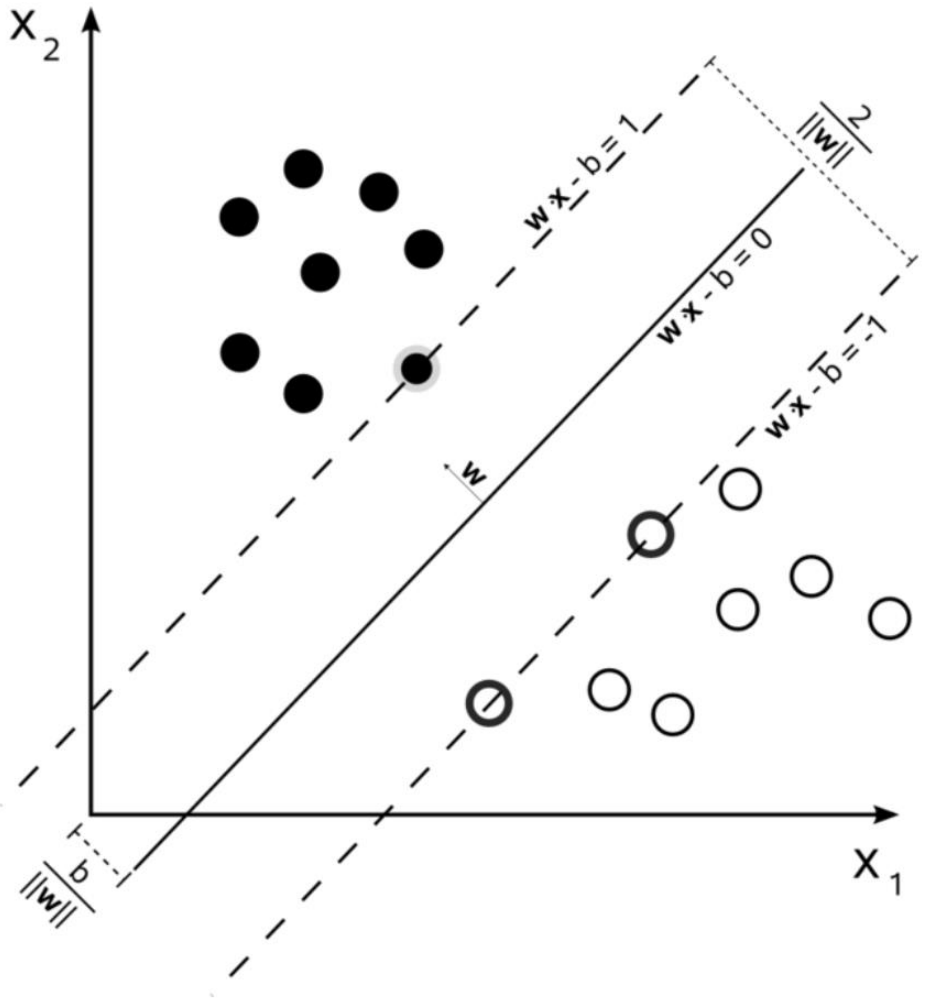
Vùng không gian giới hạn hai phần:

$$H = \begin{cases} w_i x_i + b \leq 1 \, nếu \, y_i = 1 \\ w_i x_i + b \geq -1 \, nếu \, y_i = -1 \end{cases}$$

Siêu phẳng phải tìm: $H_0 = W^T + b = 0$

Khoảng cách từ một điểm đến siêu phẳng

$$\gamma_i = \frac{|w^T x_i + b|}{||w||}$$



Hình 6. Xây dựng phân loại tối ưu SVM

Xây dựng mô hình (SVM)

1.2. Tối ưu hóa mô hình SVM

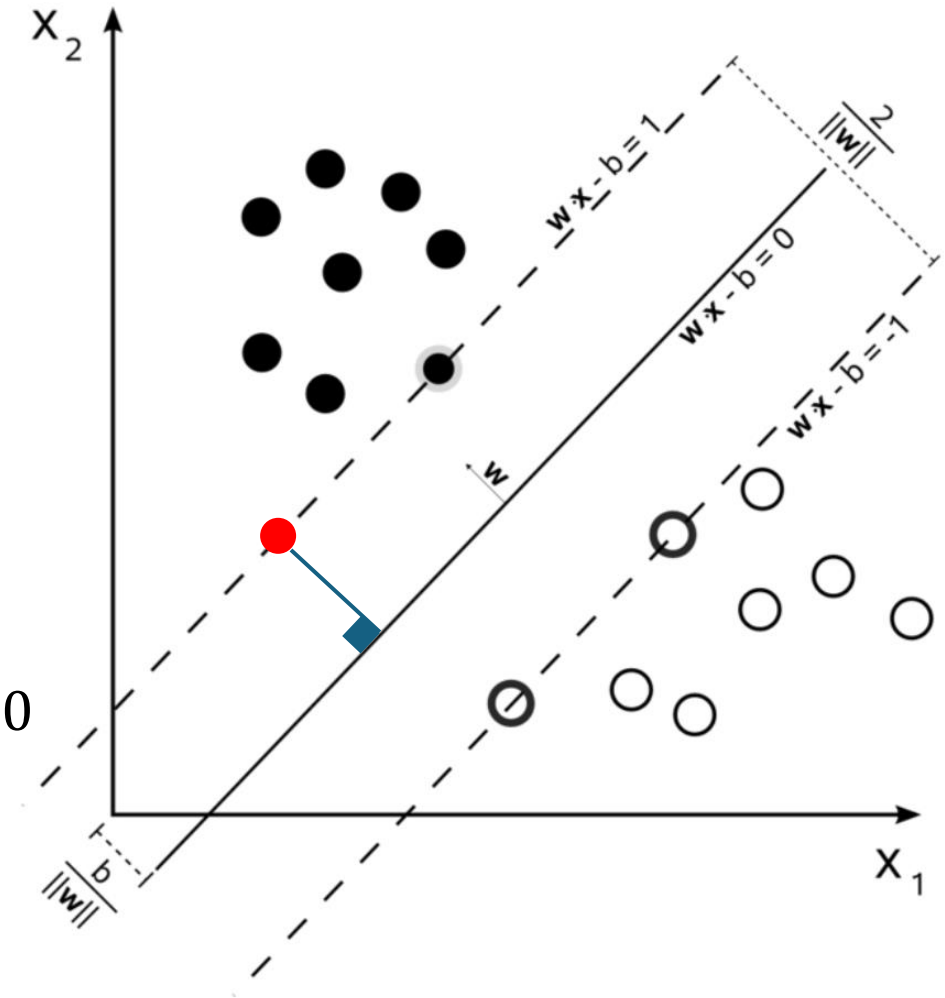
Tối ưu hóa vùng siêu phẳng

Vùng không gian H được tối đa hóa

$\frac{2(wx_0+b)}{||w||} = \frac{2}{||w||}$ max khi ||w|| min

Bài toán tối ưu của thuật toán SVM tuyến tính:

$$f(x) = \frac{1}{2} * |w^t| * |w| \rightarrow \min,\ y_i(w * x_i + b) - 1 = 0$$



Hình 6. Xây dựng phân loại tối ưu SVM

Xây dựng mô hình (SVM)

1.2. Tối ưu hóa mô hình SVM

Tối ưu hóa vùng siêu phẳng

Vùng không gian H được tối đa hóa

Kĩ thuật Langrange

$$L\big(y, f(x)\big) = \max(0, 1 - y * f(x))$$

Giải phương trình sau để cực tiểu hàm mục tiêu

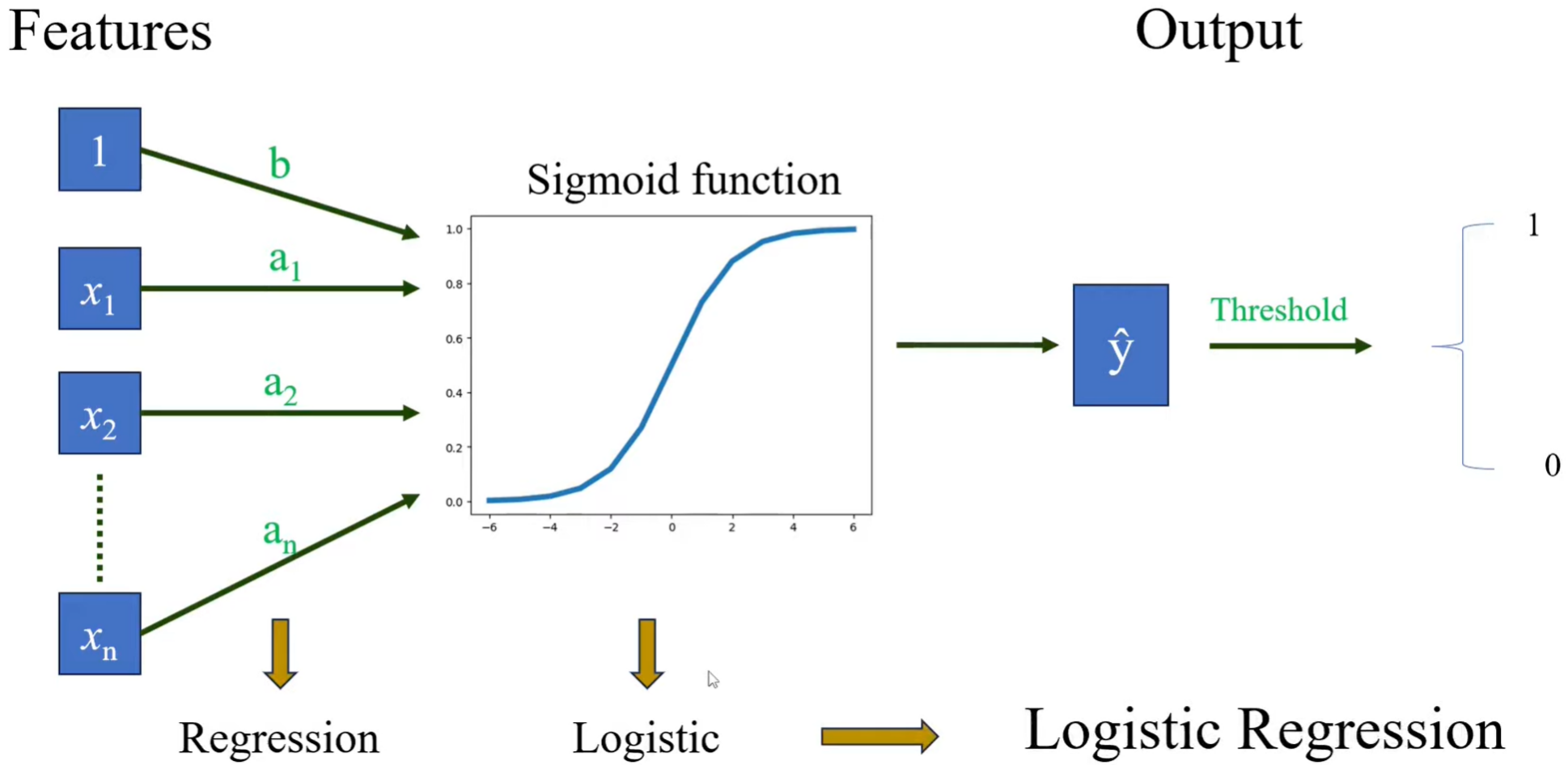
$$L(x, \lambda, \mu) = f(x) + \sum(\lambda i * gi(x)) + \sum(\mu j * hj(x)$$

Ta có nghiệm phương trình

$w = \sum_{i=1}^{N} \alpha_i y_i x_i$ và $\sum_{i-1}^{N} \alpha_i y_i = 0$

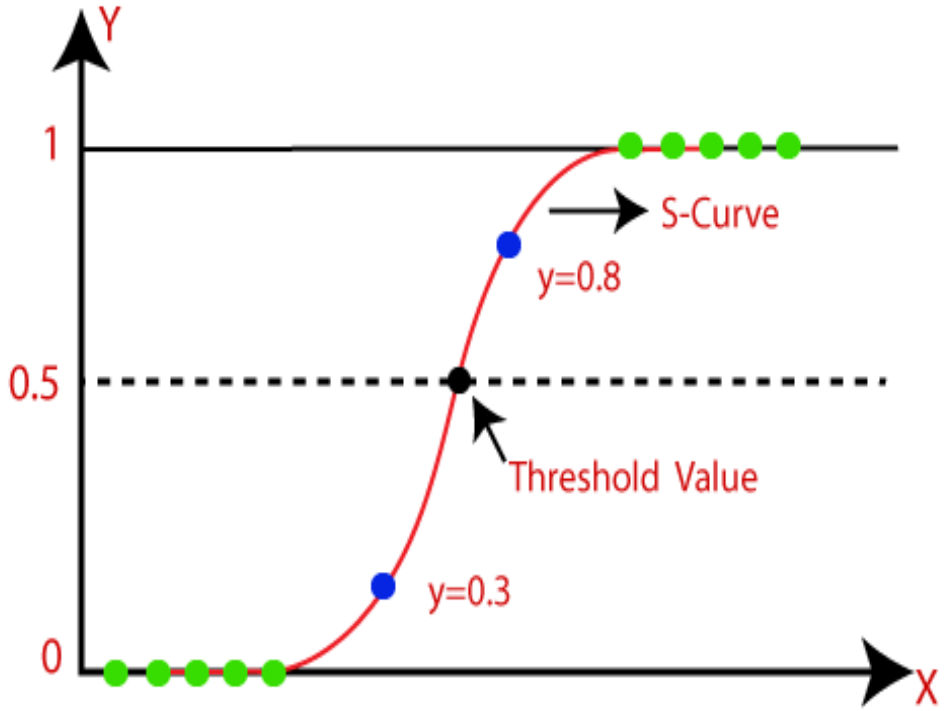
Xây dựng mô hình (Logistic Regression)

2.1. Mô hình Logistic Regression



Xây dựng mô hình (Logistic Regression)

2.1. Mô hình Logistic Regression



Hình 9. Mô hình Logistic Regression

Sigmoid function (Logistic function)

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

f(s): Hàm xác suất của biến đầu ra

$$\lim_{s \to \infty} \sigma(s) = 0; \qquad \lim_{s \to -\infty} \sigma(s) = 1$$

Xây dựng mô hình (Logistic Regression)

2.2. Hàm Loss function và tối ưu hóa mô hình Logistic Regression

Loss function

$$J(w) = -\sum_{i=1}^{N}(y_i \log(z_i) + (1 - y_i)\log(1 - z_i))$$

Tối ưu hóa mô hình

Mục tiêu của tối ưu hóa là tìm tham số w sao cho loss function được tối thiểu hóa. Ý tưởng cơ bản là di chuyển các tham số w_j theo hướng ngược lại các gradient của loss function.

Phương pháp Gradient

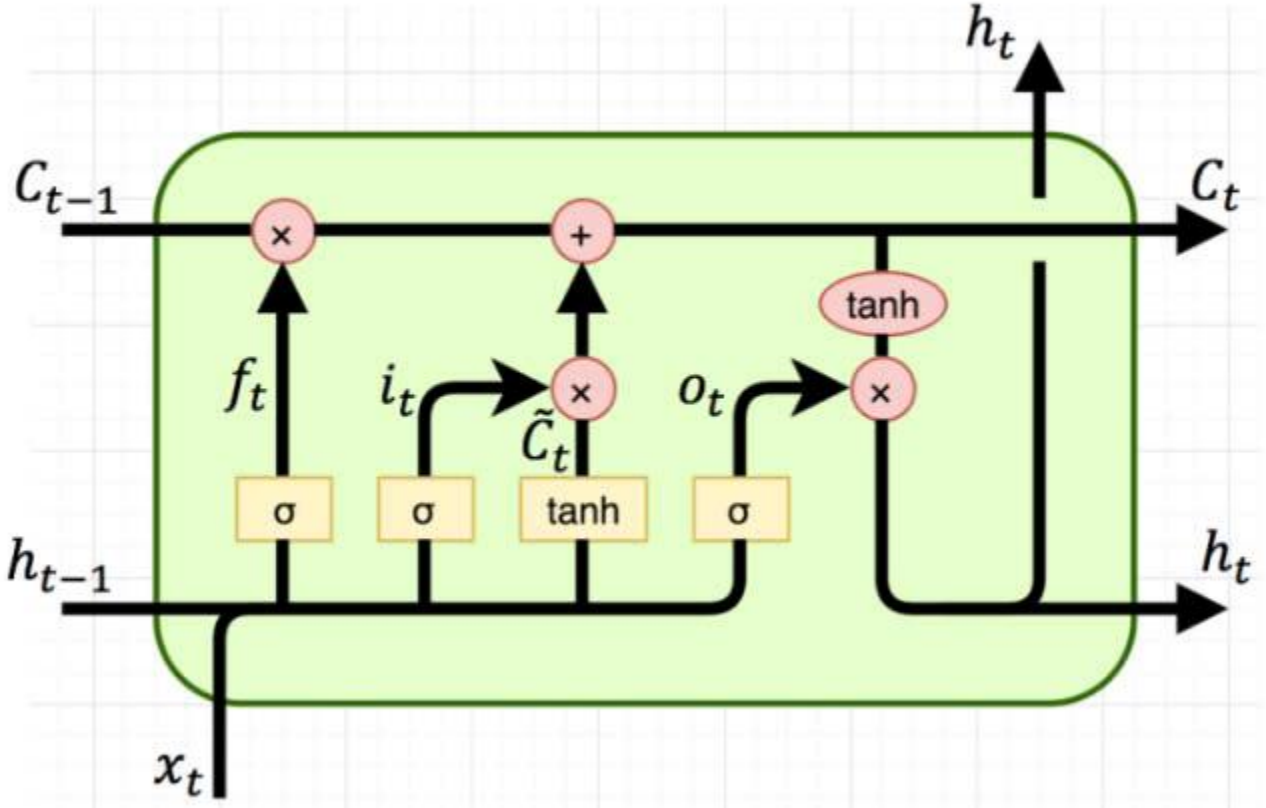
$$\frac{\partial J(w, x_i, y_i)}{\partial w} = (z_i - y_i).x$$

Khi đó ta có phương trình tối ưu hóa:

$$w_j := w_j - \alpha(z_i - y_i).x$$

Xây dựng mô hình (Long Shot-Term Memory)

3.1. Cấu trúc của LSTM



Hình 10. Mô hình LSTM

Input gate: Xác định thông tin nào từ đầu vào hiện tại nên được lưu trữ trong trạng thái bộ nhớ của LSTM

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

Tạo giá trị candidate cho trạng thái mới

$$\widetilde{C_t} = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

Cập nhật: $C_t = f_{t-1} \odot C_{t-1} + i_t \odot \widetilde{C_t}$

Forget gate: Quyết định phần nào của thông tin lưu trữ trong trạng thái bộ nhớ hiện tại nên bịquên đi.

$$f_t = \sigma\big(W_t[h_{t-1}, x_t) + b_f\big)$$

Cập nhật: $C_t = f_t \odot C_{t-1}$

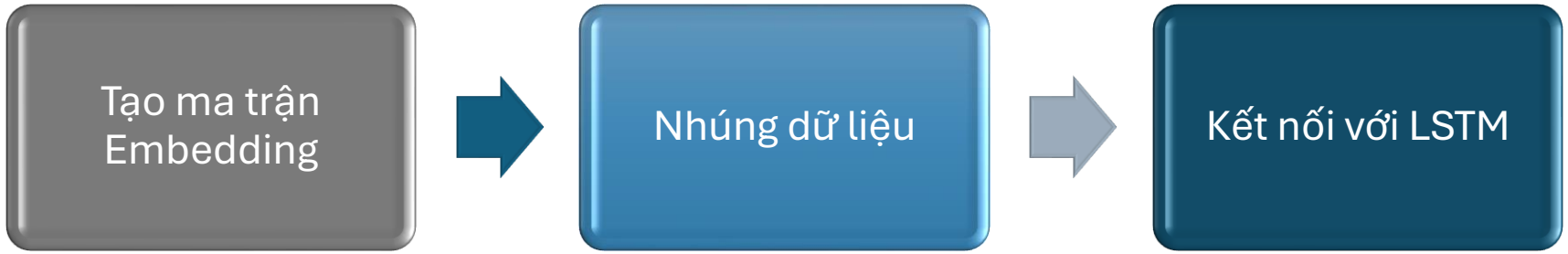
Output Gate: Quyết định phần nào của trạng thái bộ nhớ sẽ được xuất ra như là đầu ra của mạng

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_0)$$
$$h_t = o_t \odot \tanh(C_t)$$

Xây dựng mô hình (Long Shot-Term Memory)

3.2. Embedding

Embedding là một kỹ thuật dùng để ánh xạ các phần tử rời rạc của một không gian (như cáctừ trong từ vựng) vào một không gian liên tục có kích thước nhỏ hơn. Mục đích là để gán mỗi phần tử một vectơ số, giúp mô hình dễ dàng xử lý và học các quan hệ giữa chúng



Tạo ma trận có kích thước [V,D]
V:số lượng phần tử trong từ vựng.
D:kích thước của không gian nhúng

lớp embedding sử dụng ma trận đã học để chuyển đổi các chỉ số này thành các vectơ nhúng.

Các vectơ nhúng sau đó được cung cấp cho LSTM như là đầu vào. LSTM sẽ xử lý các vectơ này để học các mẫu và quan hệ trong chuỗi dữ liệu.

Hình 7. Quá trình Embedding

Xây dựng mô hình (Long Shot-Term Memory)

3.3. Dropout

Dropout là một phương pháp đơn giản nhưng hiệu quả để ngăn chặn mô hình học quá nhiều chi tiết không cần thiết từ dữ liệu huấn luyện. Trong quá trình huấn luyện, dropout "bỏ qua"một tỷ lệ ngẫu nhiên các neuron hoặc kết nối giữa các lớp trong mạng nơ-ron. Điều này có nghĩalà, trong mỗi lần lặp huấn luyện, một số neuron được "tắt" ngẫu nhiên, và chỉ một phần của môhình được sử dụng để cập nhật trọng số.

Quá trình huấn luyện

Ngẫu nhiên bỏ qua Neutron

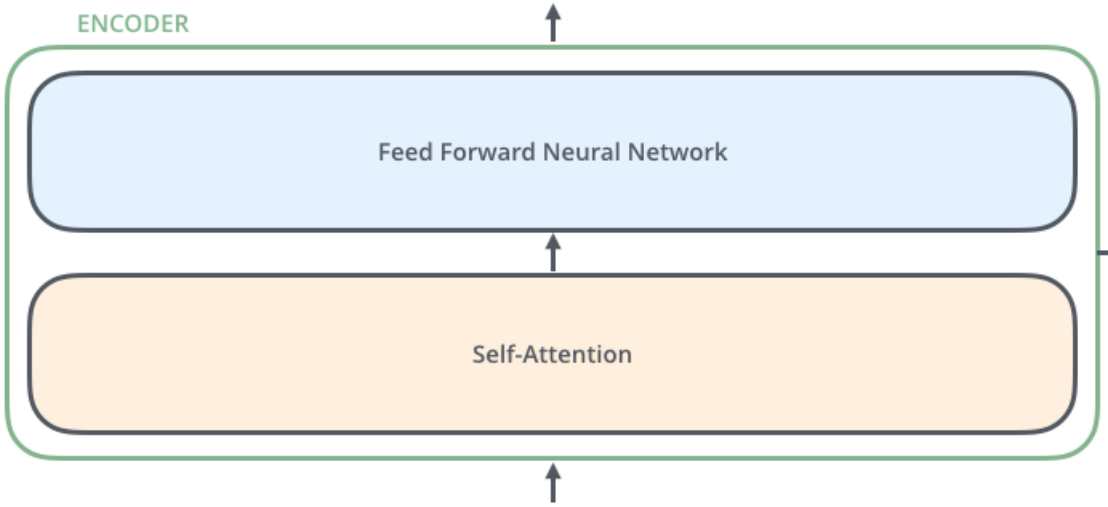
Cập nhật trọng số

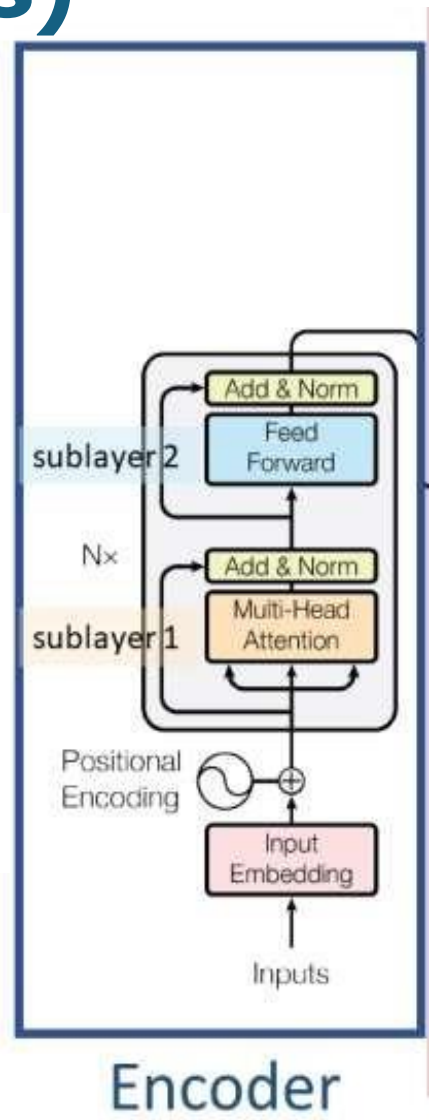
Quá trình dự đoán

Sử Dụng Tất Cả Neuron: Khi mô hình đã được huấn luyện và đang ở chế độ dự đoán (inference), tất cả các neuron sẽ được sử dụng. Để bù đắp cho việc dropouttrong quá trình huấn luyện, các trọng số của các neuron sẽ được điều chỉnh bằng cách nhân với tỷ lệ không bị dropout (1 – p)

Xây dựng mô hình (Transformers)

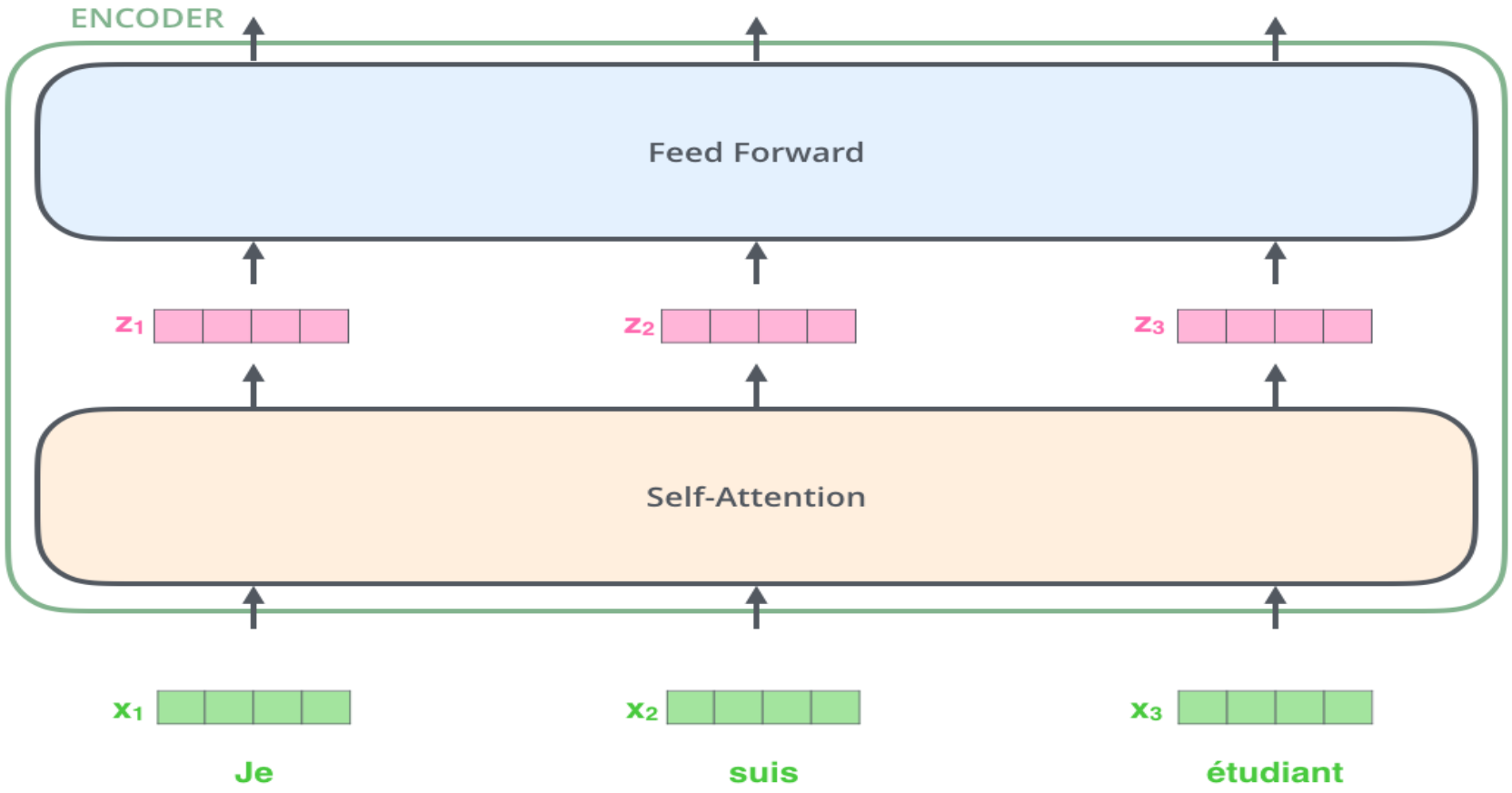
4.1. Encoder





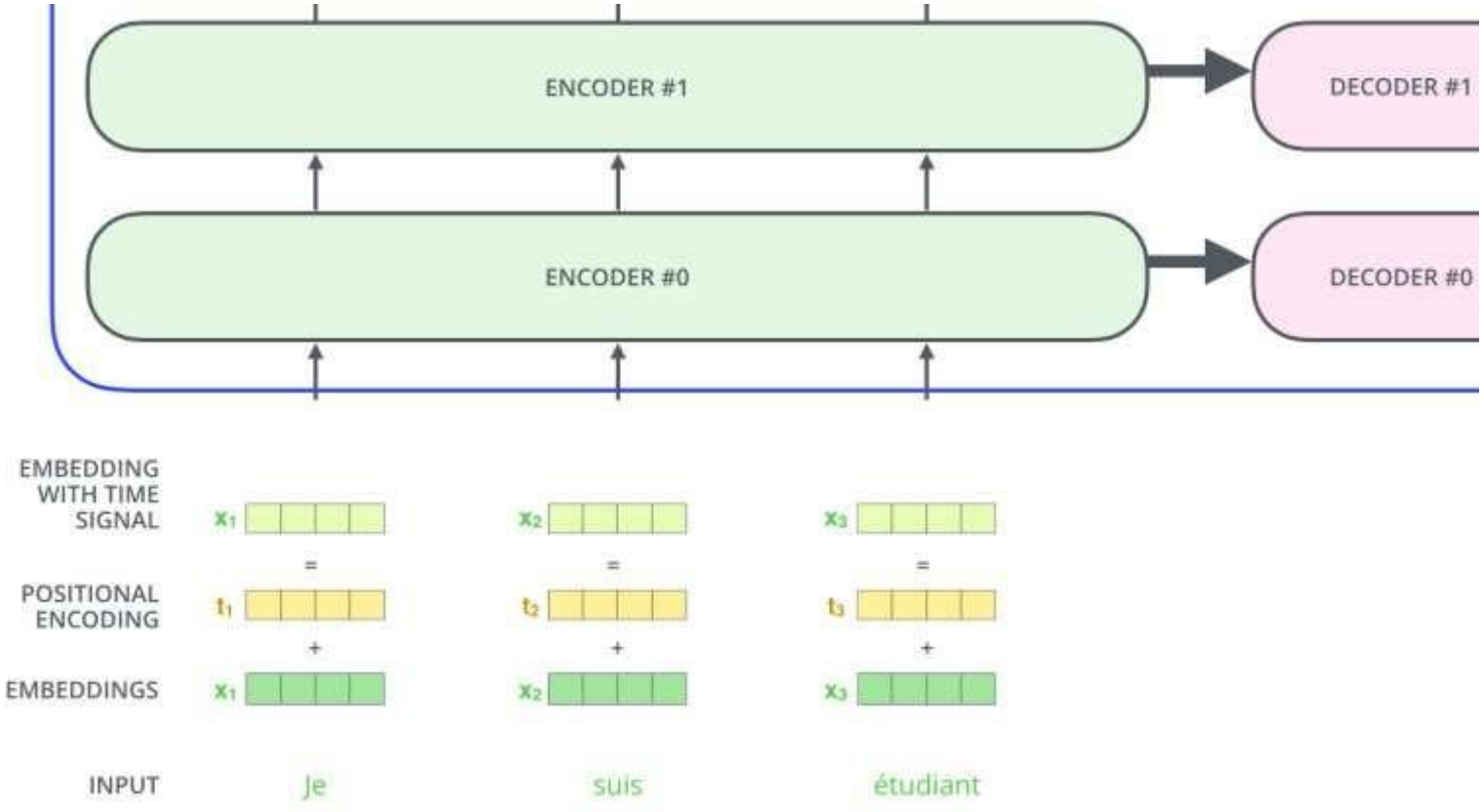
Xây dựng mô hình (Transformers)

4.1. Encoder



Xây dựng mô hình (Transformers)

4.2. Positional Encoding



- Tất cả các vector được đưa vào mạng cùng 1 lúc, song song
- Cần một cơ chế để ”note” lại vị trí các từ trong câu chính là Positional Encoding

Xây dựng mô hình (Transformers)

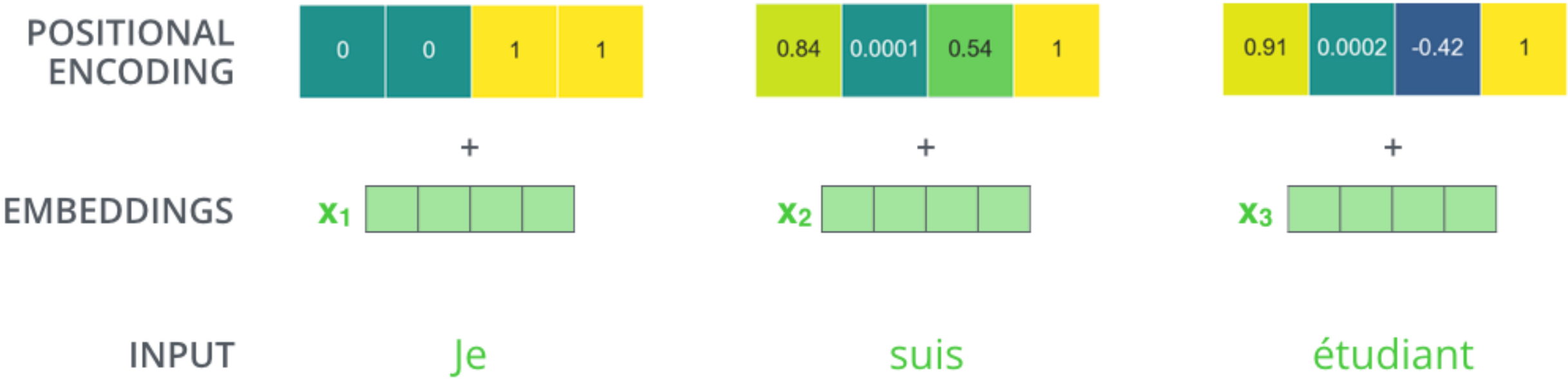
4.2. Positional Encoding

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

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

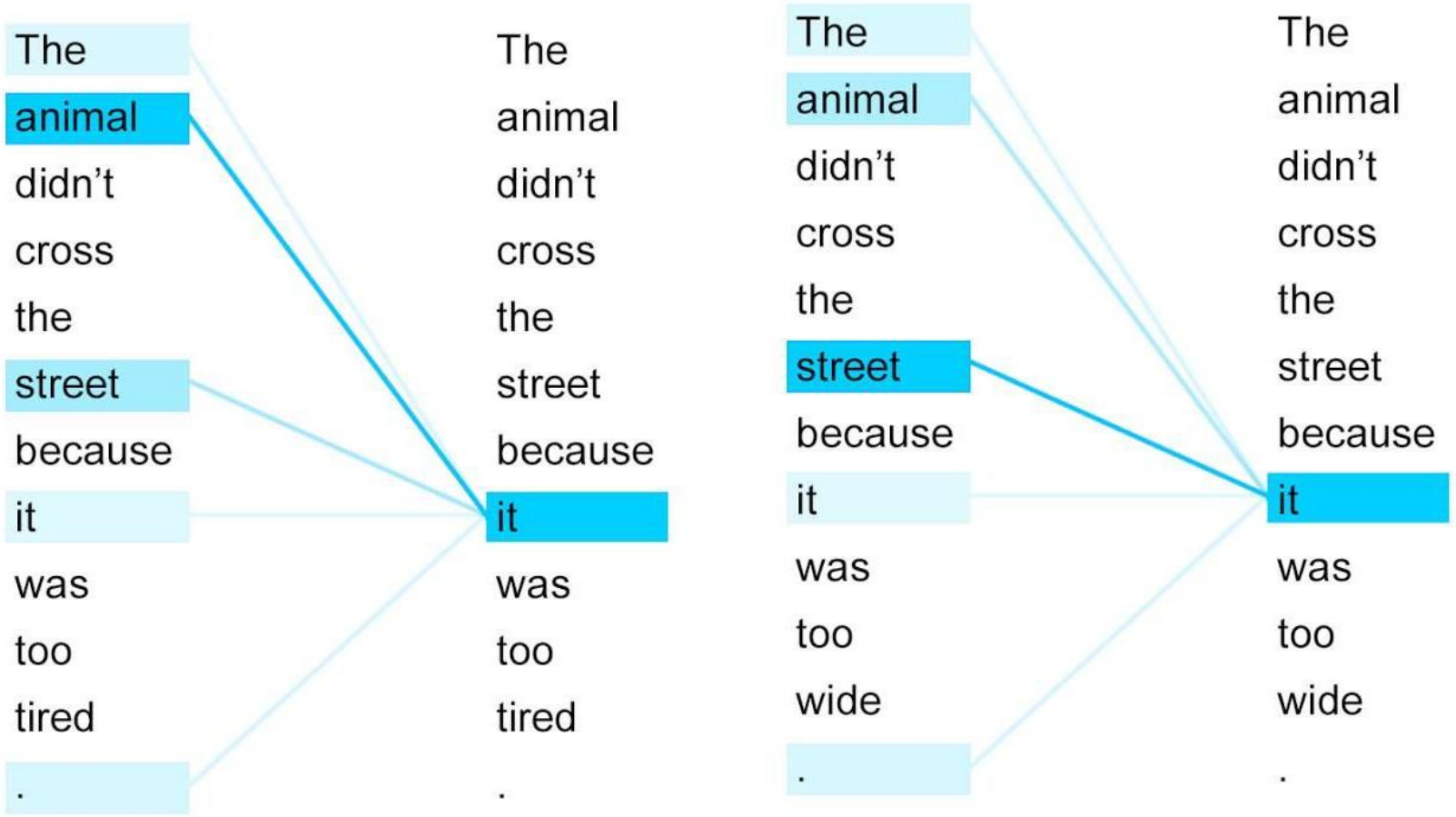
$$p_{i,j} = \begin{cases} sin\left(\frac{i}{10000^{\frac{j-1}{d_{emb\_dim}}}}\right) \\ \cos\left(\frac{i}{10000^{\frac{j-1}{d_{emb\_dim}}}}\right) \end{cases}$$

- Có thể dùng idex but…
- Tác giả dùng hàm sin, cosin



Xây dựng mô hình (Transformers)

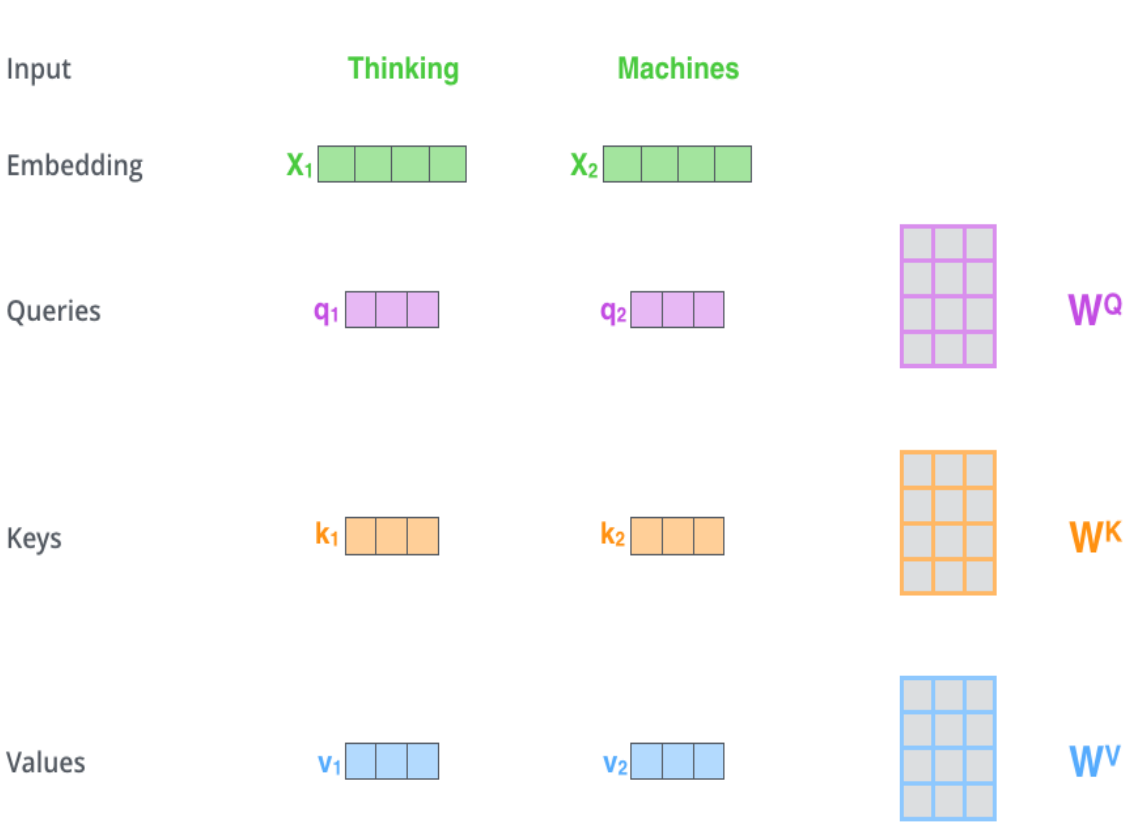
4.2. Positional Encoding

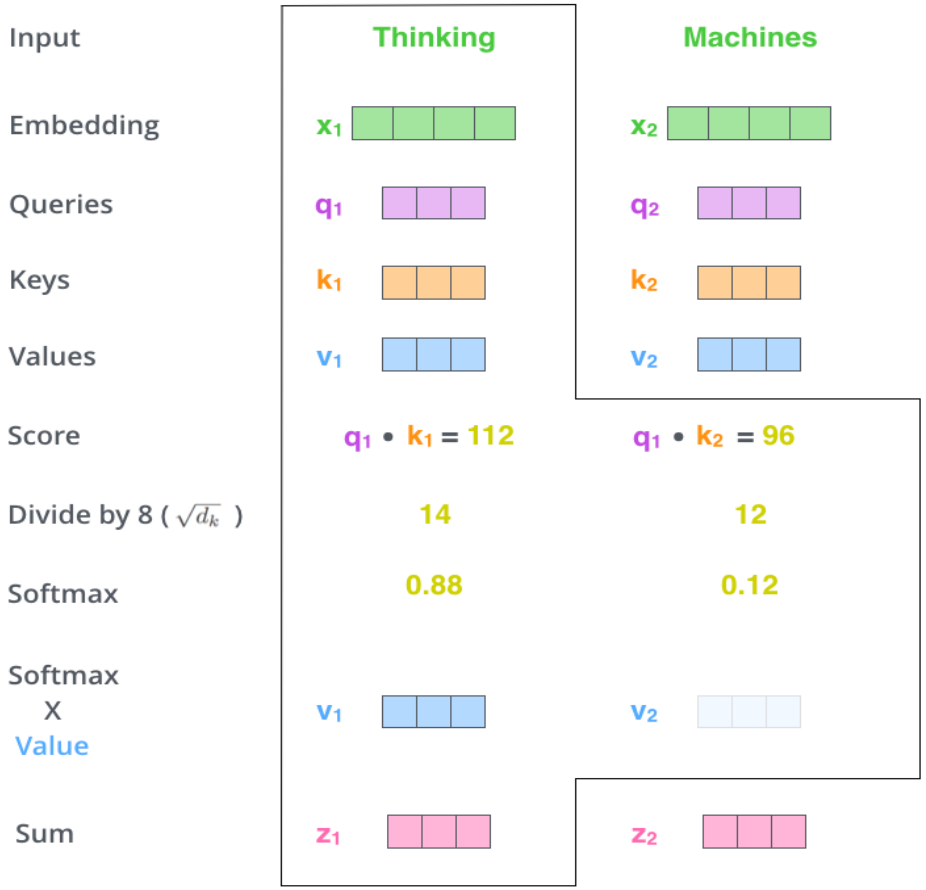


- Tạo ra quan hệ giữa các từ trong câu
- Khi được mã hoá (encode) nó sẽ mang thêm thông tin của các từ liên quan

Xây dựng mô hình (Transformers)

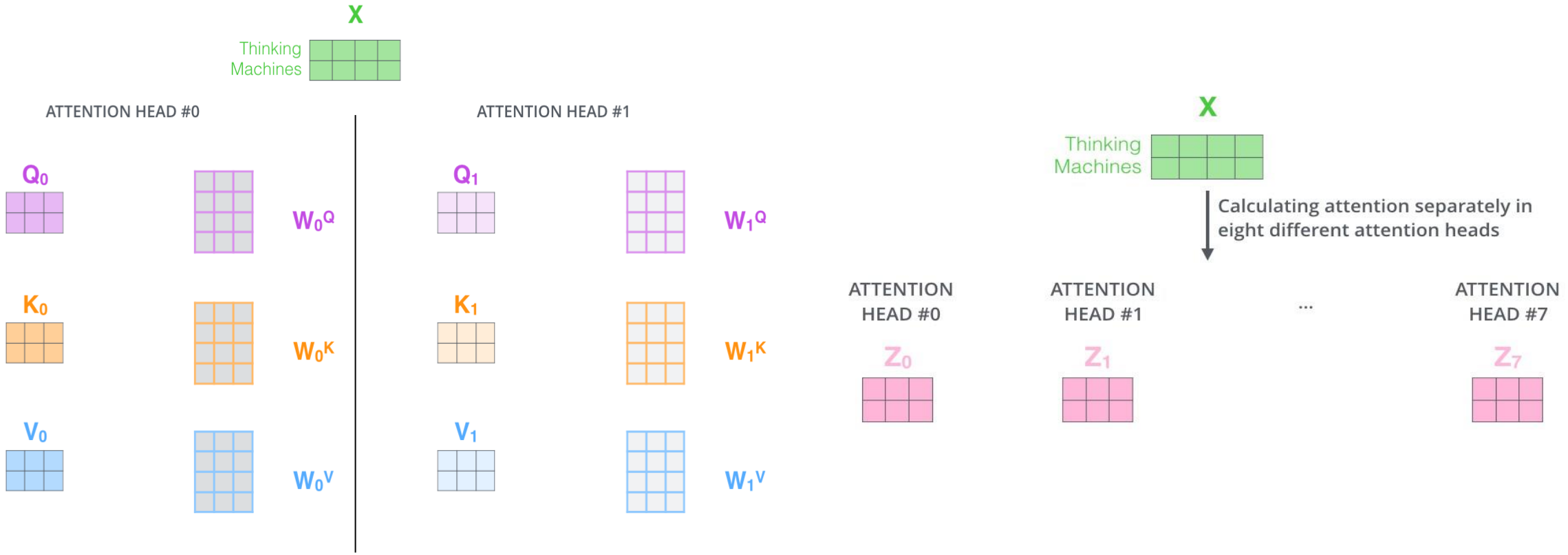
4.3. Self Attention





Xây dựng mô hình (Transformers)

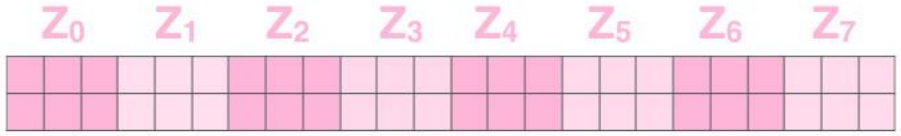
4.4. Multi-head



Xây dựng mô hình (Transformers)

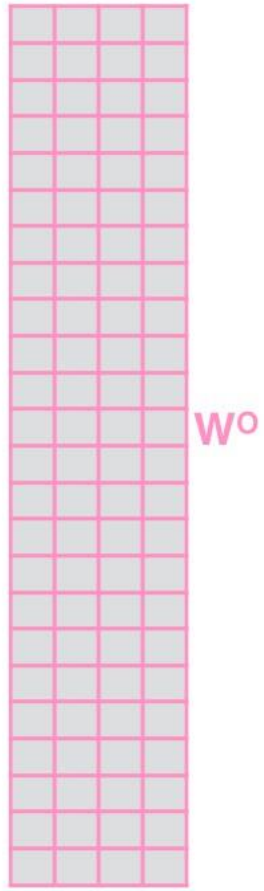
4.4. Multi-head

1) Concatenate all the attention heads



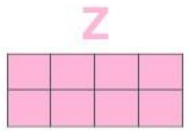
2) Multiply with a weight matrix W^O that was trained jointly with the model

X



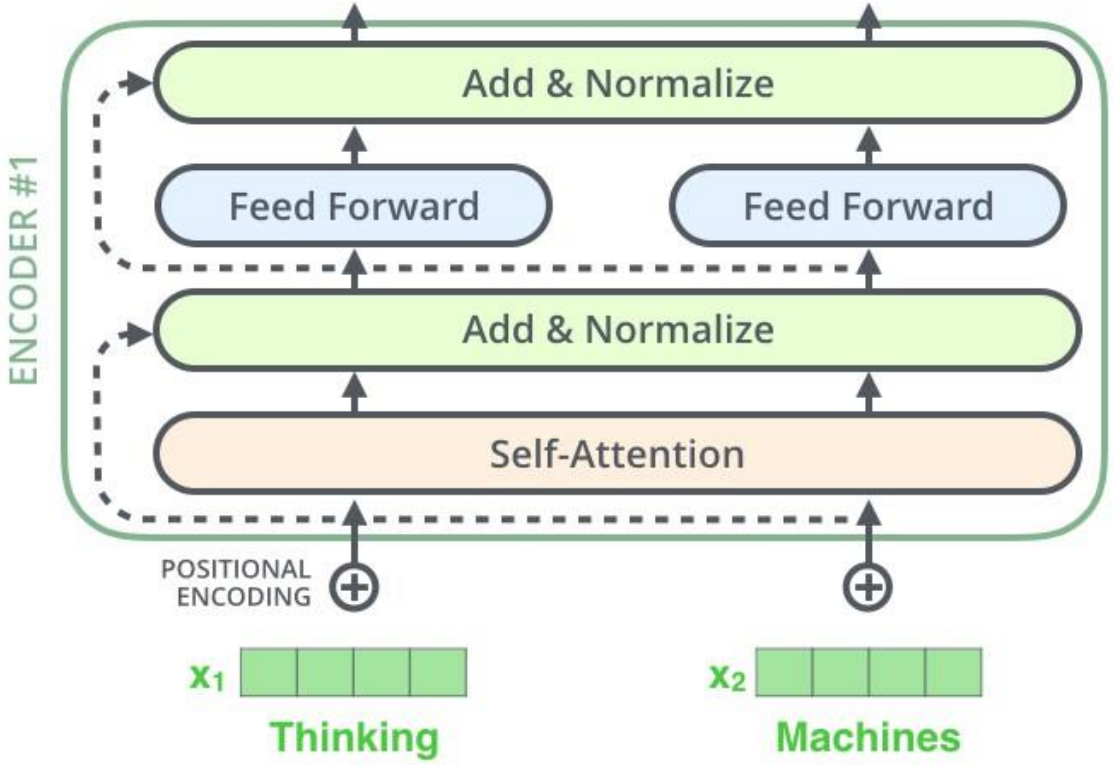
3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

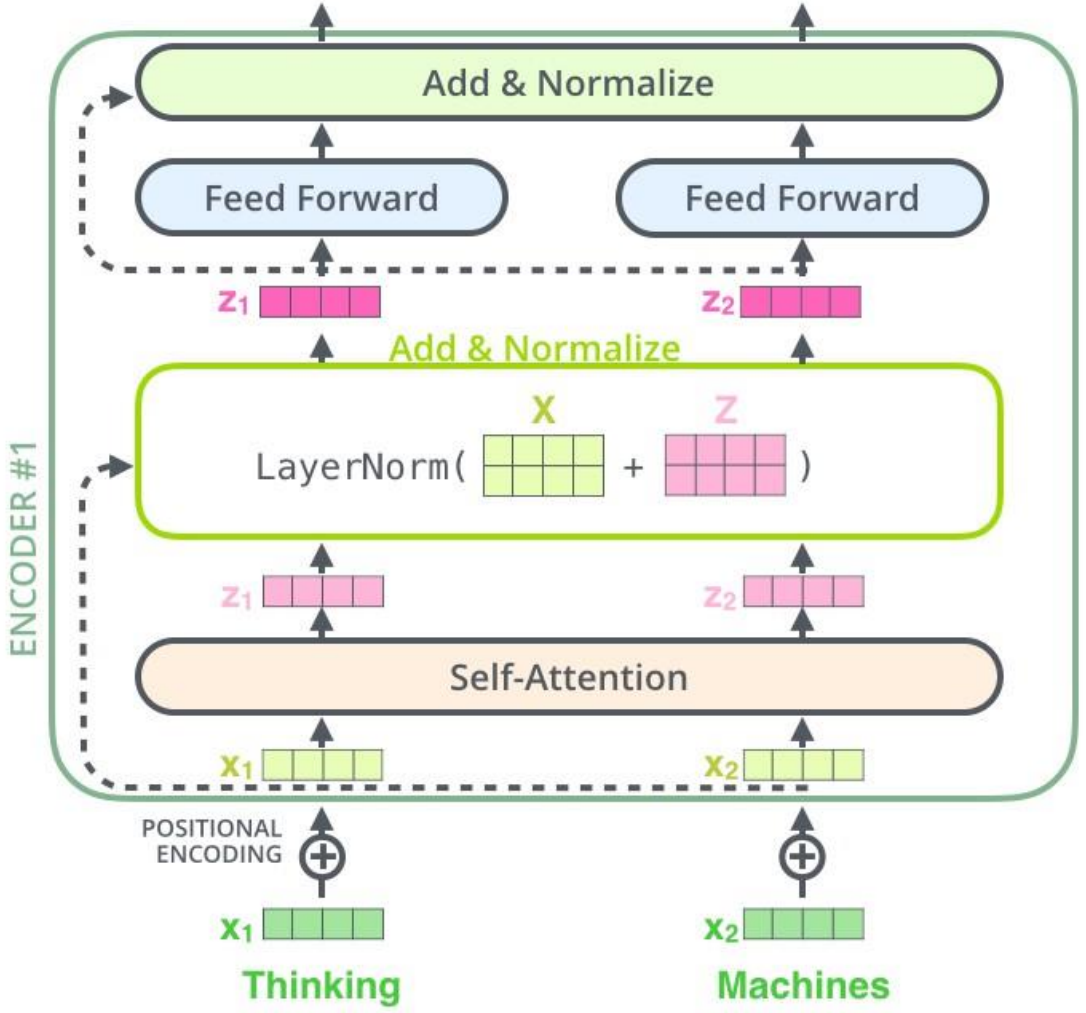
=



Xây dựng mô hình (Transformers)

4.5. The residual





Kết quả

1. Kết quả SVM và Logistic Regression

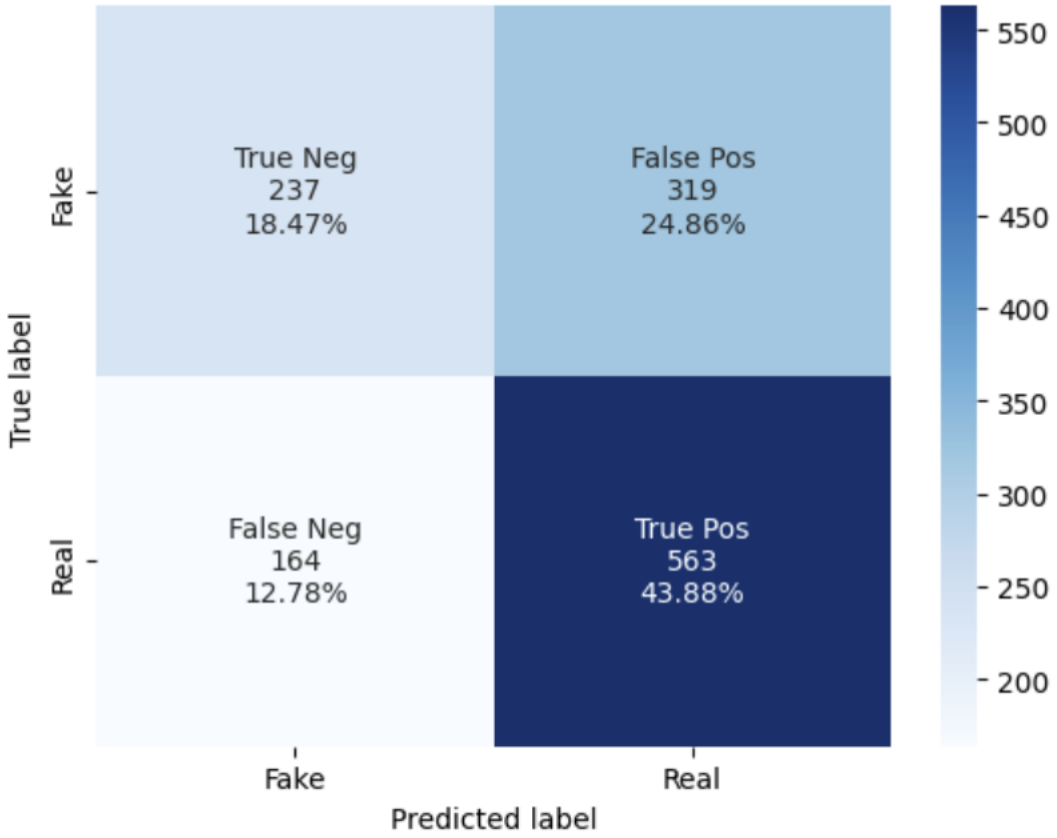
	Precision				Recall				F1-score			
	SVM-CV	SVM-TF IDF	LR-CV	LR-TF IDF	SVM-CV	SVM-TF IDF	LR-CV	LR-TF IDF	SVM-CV	SVM TF IDF	LR-CV	LR-TF IDF
False	0.59	0.63	0.56	0.41	0.43	0.38	0.51	0.41	0.5	0.48	0.53	0.48
True	0.64	0.64	0.65	0.78	0.77	0.83	0.69	0.78	0.7	0.72	0.67	0.7
Acc									0.62	0.64	0.61	0.62
Macro avg	0.61	0.63	0.6	0.59	0.6	0.61	0.6	0.5	0.6	0.6	0.6	0.59
Weighted	0.62	0.63	0.61	0.62	0.61	0.64	0.61	0.62	0.61	0.61	0.61	0.61

Hình 1.1. Kết quả mô hình SVM và Logistic Regression sử dụng phương pháp CountVectorizer (CV) và TF-IDF (TF IDF) làm vecto đặc trưng

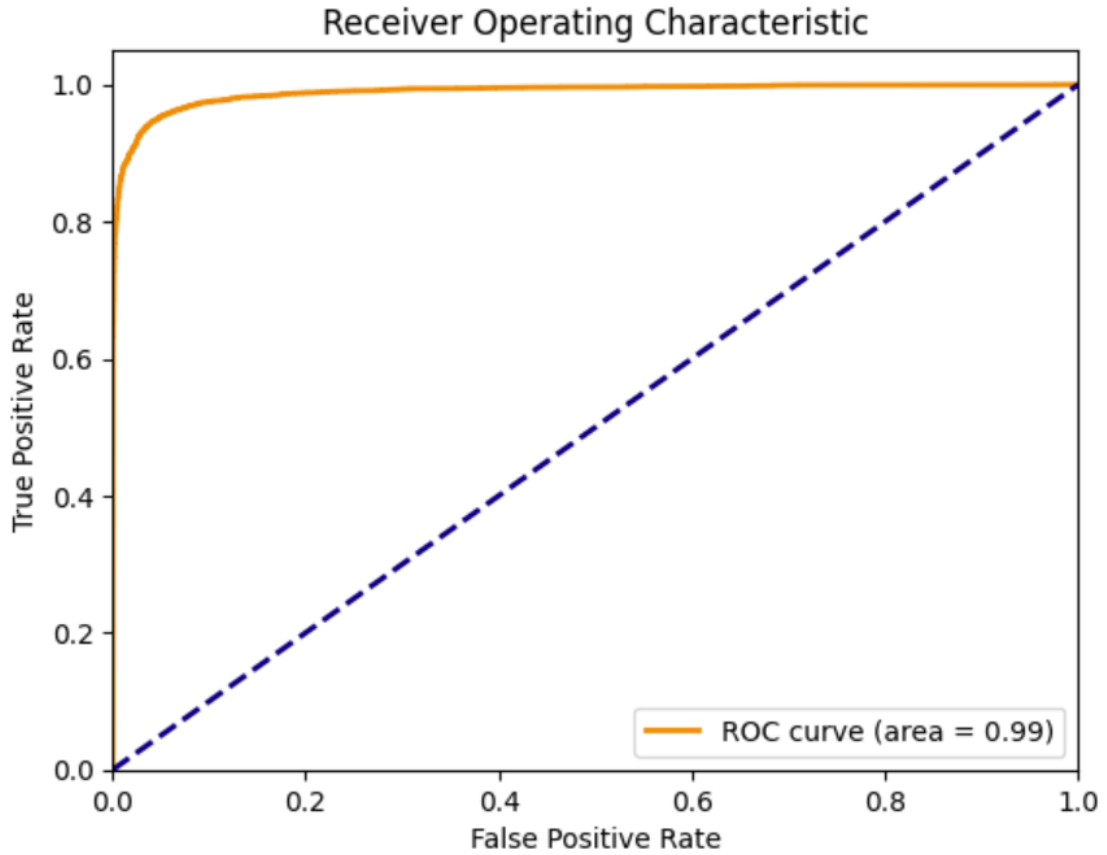
Kết quả

1. Kết quả SVM và Logistic Regression

1.1. Mô hình kết quả SVM



Hình 1.2. Confusion matrix for SVM

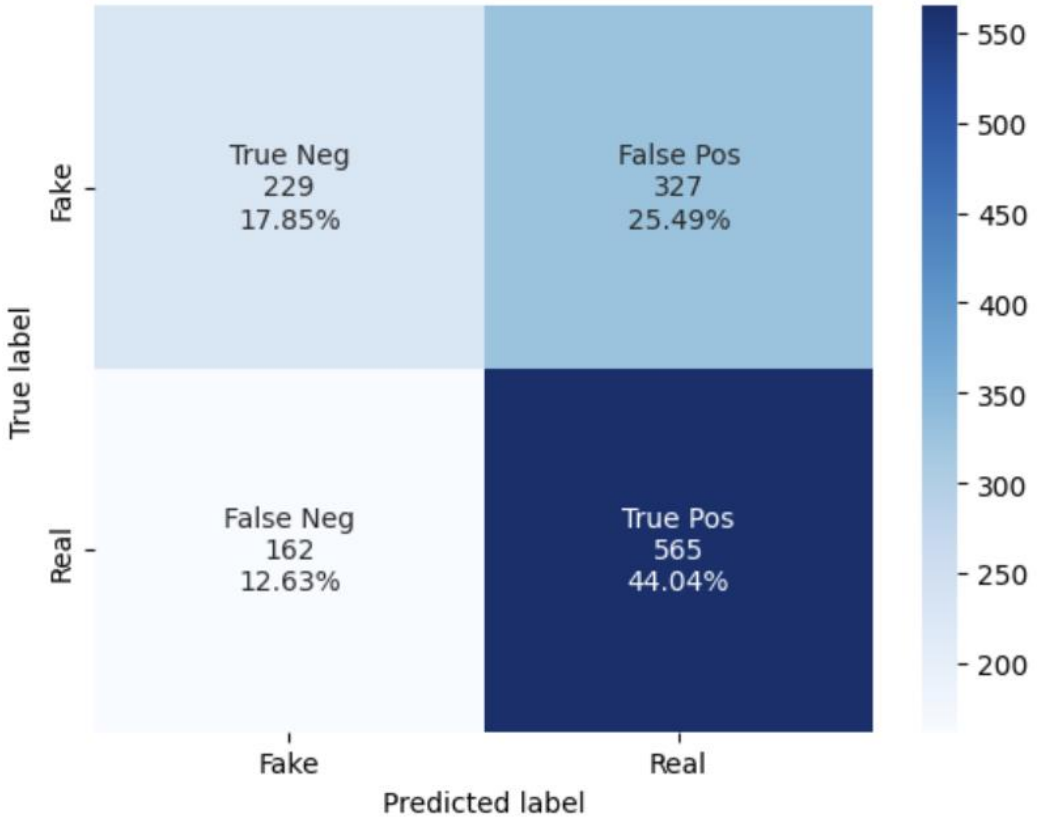


Hình 1.3. Biểu đồ ROC của SVM

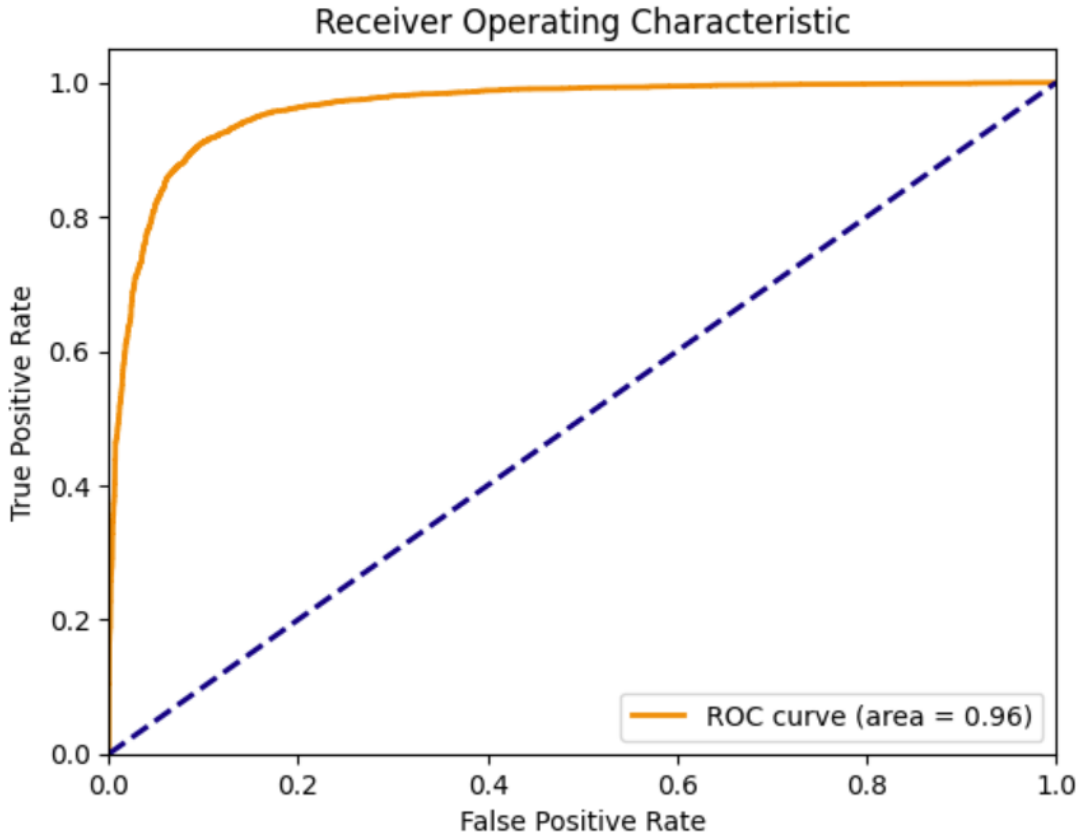
Kết quả

1. Kết quả SVM và Logistic Regression

1.2. Mô hình kết quả Logistic regression



Hình 2.2. Confusion matrix for Logistic Regression



Hình 2.3. Biểu đồ ROC của Logistic Regression

Kết quả

2. Kết quả mô hình LSTM

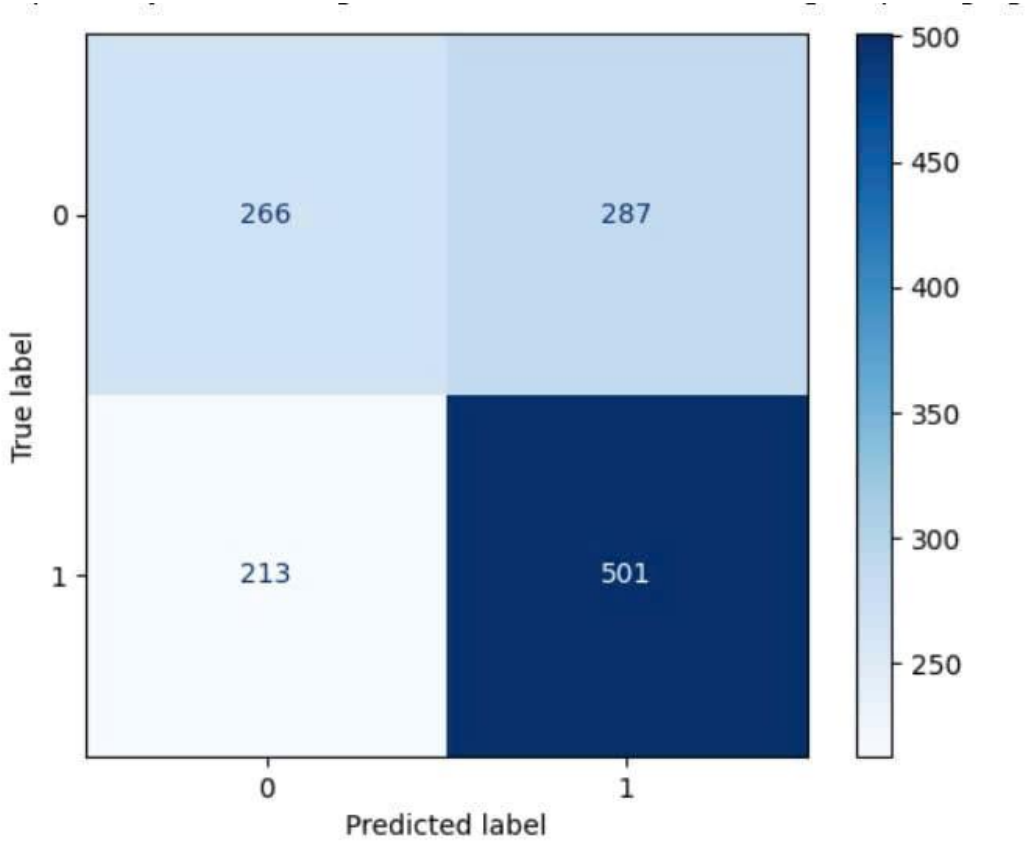
Classification Report:

	precision	recall	f1-score	support
False	0.56	0.48	0.52	553
True	0.64	0.70	0.67	714
accuracy			0.61	1267
macro avg	0.60	0.59	0.59	1267
weighted avg	0.60	0.61	0.60	1267

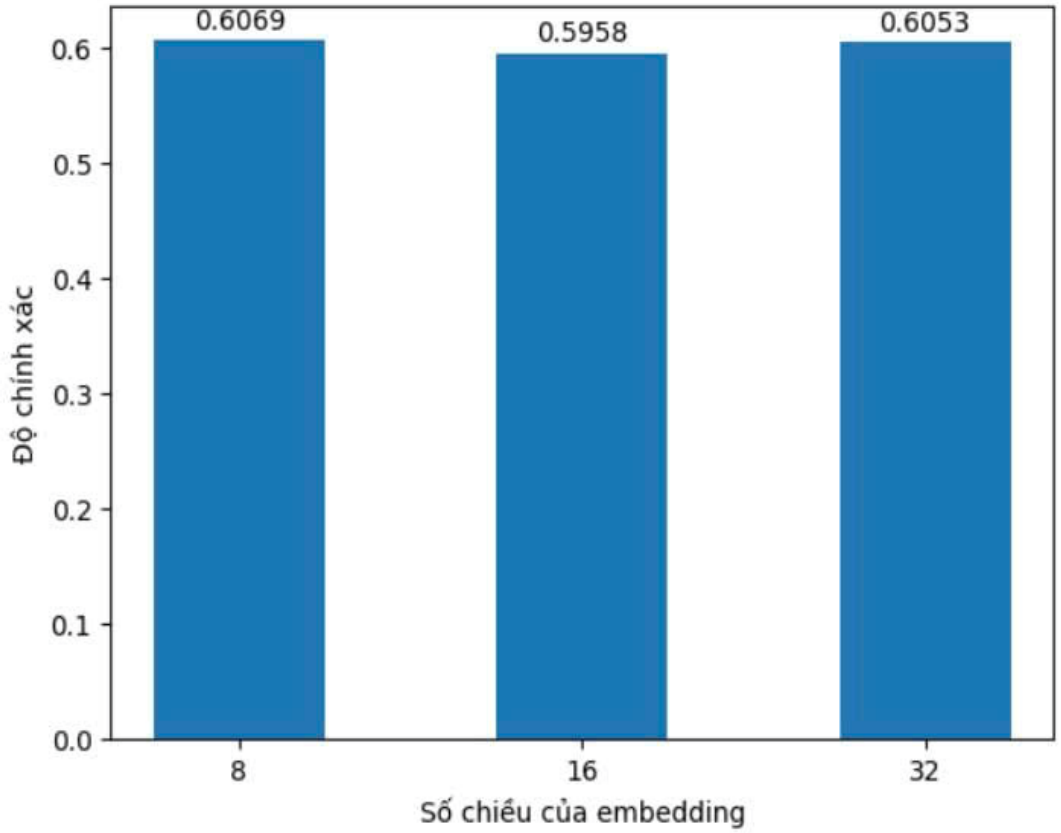
Hình 3.1. Kết quả mô hình LSTM

Kết quả

2. Kết quả mô hình LSTM



Hình 3.2. Confusion matrix for LSTM



Hình 3.3. Biểu đồ độ chính xác với các số chiều khác nhau của lớp Embedding

Kết quả

3. Kết quả mô hình Transformers

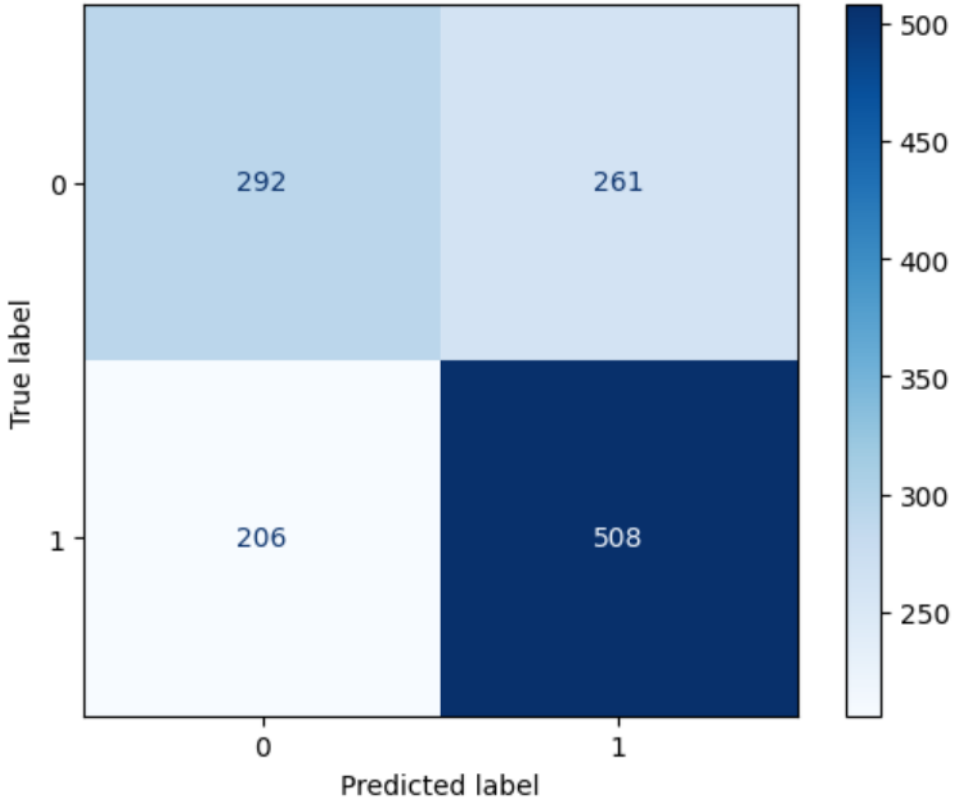
Classification Report:

	precision	recall	f1-score	support
false	0.59	0.46	0.52	553
true	0.64	0.76	0.69	714
accuracy			0.63	1267
macro avg	0.62	0.61	0.61	1267
weighted avg	0.62	0.63	0.62	1267

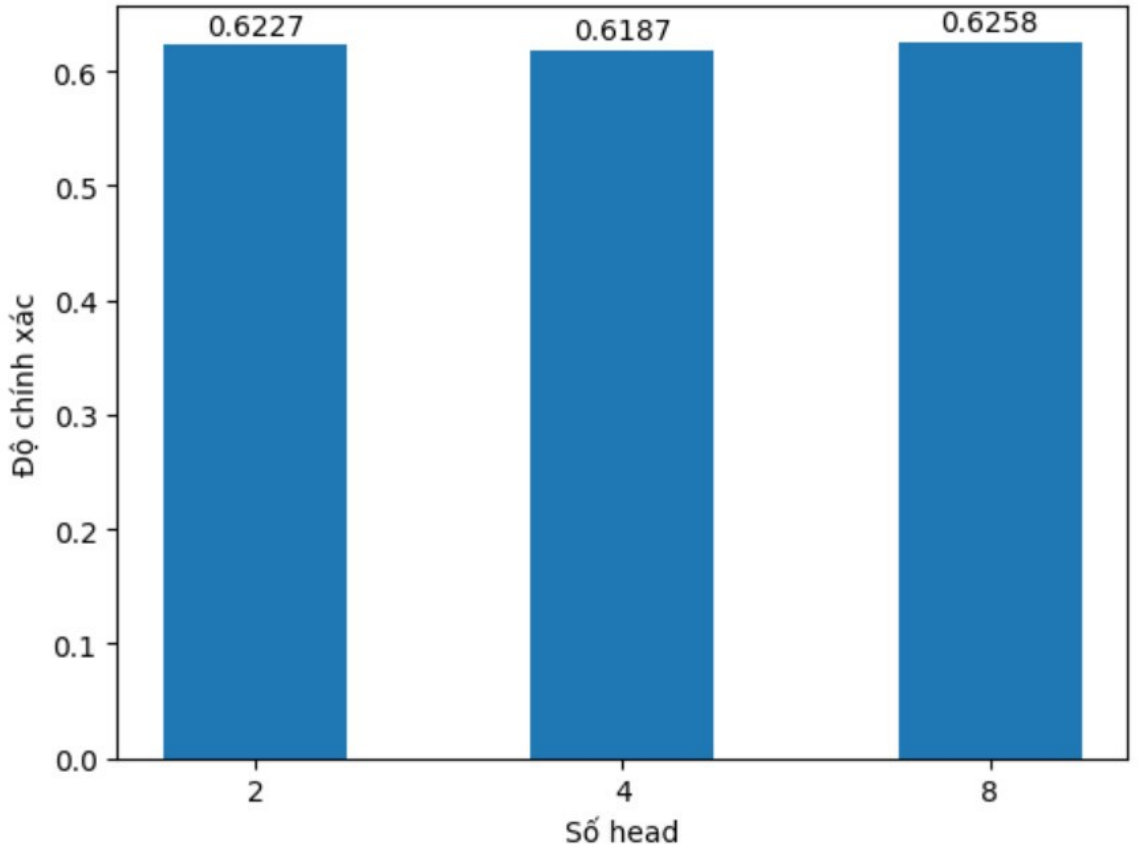
Hình 4.1 Kết quả mô hình Transformers

Kết quả

3. Kết quả mô hình Transformers



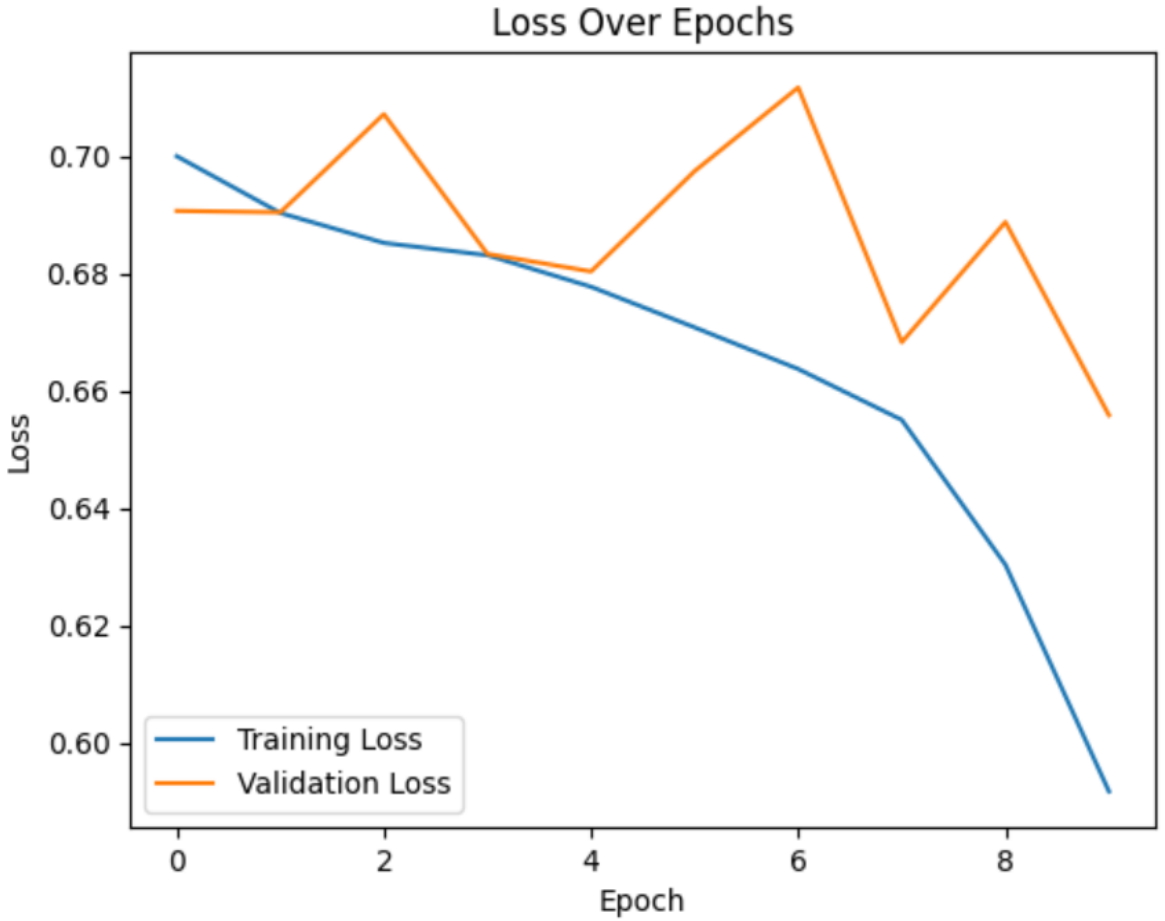
Hình 4.2. Confusion matrix for Transformers



Hình 4.3. Biểu đồ độ chính xác với số head khác nhau ở lớp Multi-head Attention

Kết quả

3. Kết quả mô hình Transformers



Hình 4.4. Biểu đồ hàm mất mát

Hình 4.5. Biểu đồ độ chính xác

Kết luận

Theo hướng truyền thống học máy

Ưu điểm

Các mô hình ML truyền thống thường hoạt động tốt trên các tập dữ liệu không quá lớn và có tính toán nhanh hơn so với các mô hình học sâu

- **Logistic Regression** có thể cho kết quả tốt khi kết hợp với các kỹ thuật xử lý văn bản như TF-IDF hoặc các bộ biểu diễn từ (word embeddings).
- **SVM:** Tránh được overfitting

Nhược điểm

Khả năng mở rộng kém
Hiệu suất hạn chế
Khả năng mô hình hóa còn hạn chế

Theo hướng học sâu

Ưu điểm

LSTM: Giải quyết vấn đề biến mất và bùng nổ gradient
Xử lí thông tin tuần tự dài hạn

Transformers:
Hiệu suất cao
Khả năng hóa song song
Mô hình hóa sự phụ thuộc dài hạn tốt

Nhược điểm

LSTM: Tính toán phức tạp, khó khan trong việc mô hình hóa sự phụ thuộc dài hạn

Transformers: Đòi hỏi tài nguyên lớn
Khó huấn luyện

Tài liệu tham khảo

[1] Berwick, R. An idiot's guide to support vector machines (svms). Retrieved on October 21 (2003), 2011.

[2] Bishop, C. M. Pattern recognition and machine learning. Springer google schola 2 (2006), 1122–1128.

[3] Hùng, V. T., Chi, N. K., and Kiệt, T. A. Phát hiện tự động tin giả: Thành tựu và thách thức. Tạp chí Khoa học và Công nghệ-Đại học Đà Nẵng (2022), 71–78.

[4] Yacouby, R., and Axman, D. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems (2020), pp. 79–91.

Thank you for listening

Hanoi University of Science

VNU university