FPT Education

FPT UNIVERSITY

XBERT - MODEL FOR HATE SPEECH DETECTION IN VIETNAMESE

INTRODUCTION

In the digital age, social media's pervasive influence has inadvertently escalated the prevalence of hate speech and offensive comments, with alarming implications for mental health.

Addressing this critical issue, our research introduces XBert, a model for detecting hostile and provocative language in Vietnamese.

INTRODUCTION

Trung Quân Idol bị trầm cảm 2 năm chỉ vì đọc bình luận của khán giả

SAO VIỆT | Thứ Sáu, 11/08/2023 06:57:31 +07:00 Theo dõi VTC NEWS trên Google News

(VTC News) - Trung Quân Idol lần đầu có những chia sẻ về câu chuyện áp lực tâm lý khi hoạt động nghệ thuật cùng đàn chị Thu Minh.

VNEXPRESS | Chủ nhật, 3/12/2023 | TP HCM 29 ° | Mới nhất | Tin theo khu vực

Góc nhìn Thế giới Video Podcasts Kinh doanh Bất động sản Khoa học Giải trí Thể thao Pháp luật Giáo dục Sức kh

Ý kiến — Thứ ba, 15/10/2019, 09:16 (GMT+7)

Bình luận mạng xã hội - 'thòng lọng' treo trên đầu mỗi người

Nhiều năm về trước, khái niệm trầm cảm không hề phổ biến nhưng hiện nay, chúng ta lại nghe đến căn bệnh này hàng ngày.

Ngày 14/10, nữ nghệ sỹ Choi Jin Ri (nghệ danh Sulli) qua đời tại nhà riêng. Hầu hết người hâm mộ cô ấy đều bàng hoàng. Tôi cũng vậy, mặc dù tôi không thần tượng

PRERPROCESSING

NORMALIZE

òa	oà
Òa	Oà
ÒA	OÀ
òe	oè
ùy	uỳ
Ủy	Uỷ
ỦY	UỶ

REMOVE EMOJI

PRERPROCESSING

TEENCODE DECODE

k	không
ko	không
kh	không
đc	được
dc	được
vs	với
v	vậy

Easy Data Augmentation Techniques

Easy Data Augmentation Techniques (EDA) are a set of simple and efficient methods designed to enhance machine learning models by increasing the size and diversity of training datasets. These techniques include operations like synonym replacement, random insertion, random swap, and random deletion.

Synonym Replacement (SR)

Randomly choose n words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.

Câu gốc	Nhân viên nhiệt tình và lịch sự
Synonym Replacement (SR)	Nhân viên hoan nghênh và lịch sự

Random Insertion (RI)

Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this n times.

Câu gốc	Nhân viên nhiệt tình và lịch sự
Random Insertion (RI)	Nhân viên nhiệt tình và lịch sự chuyên viên cao cấp

Random Swap (RS)

Randomly choose two words in the sentence and swap their positions. Do this n times

Câu gốc	Nhân viên nhiệt tình và lịch sự
Random Swap (RS)	Nhân viên nhiệt tình và lịch sự chuyên viên cao cấp

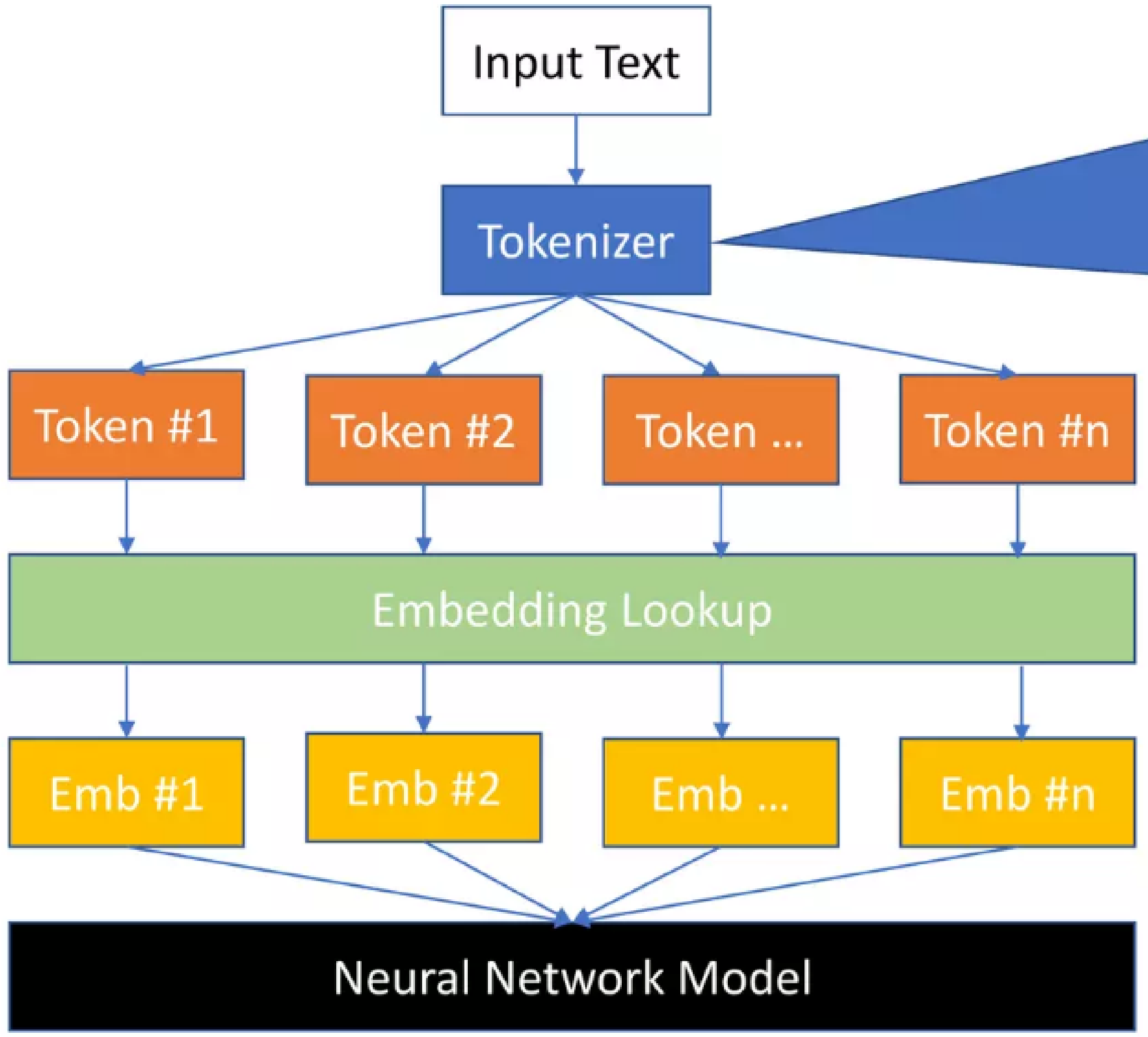
Random Deletion (RD)

Randomly remove each word in the sentence with probability p.

Câu gốc	Nhân viên nhiệt tình và lịch sự
Random Deletion (RD)	Nhân viên nhiệt tình và lịch sự chuyên viên cao cấp

Dataset	Label	Original dataset			Augmented dataset		
		Num comments	Avg word length	Vocab size	Num comments	Avg word length	Vocab size
Vi-HSD	CLEAN	19,886	6.55	130,238	19,886	6.55	130,238
	OFFENSIVE	1,606	7.24	11,624	10,147	7.57	76,802
	HATE	2,556	12.08	30,883	16,849	11.64	196,086
VLSP	CLEAN	18,614	14.85	276,557	18,614	14.85	276,557
	OFFENSIVE	1,022	8.87	9,063	8,461	8.05	68,093
	HATE	709	14.23	10,087	6,392	13.41	85,713

Tokenizer



Different levels of tokenization

1) Words
2) Characters
3) Words & Characters
4) Subwords

Word-level tokenization

This is the most commonly used tokenization technique. It splits a piece of text into words based on a delimiter. The most commonly used delimiter is space. You can also split your text using more than one delimiter, like space and punctuation marks

Language	Original Sentence	After tokenization
English	How are you	how | are | you
Bahasa	Apa kabar	apa | kabar
French	Comment allez-vous	Comment | allez-vous
Thai	คุณเป็นอย่างไรบ้าง	คุณ | เป็น | อย่างไร | บ้าง
Chinese	你好吗	你 | 好 | 吗
Japanese	お元気ですか	お | 元気 | です | か
Arabic	كيف حالك	كيف | حالك

ISSUES

The limitation of this technique is that it leads to a massive corpus and a large vocabulary, making the model more cumbersome and requiring more computational resources. To address this issue, researchers have introduced character-based tokenization techniques.

Character-Level tokenization

Character-based encoding splits raw text into individual characters. The logic behind this encoding is that a language may have many different words but a fixed number of characters. This results in a very small vocabulary size. For example, English has 256 different characters (letters, numbers, special symbols) while containing nearly 170,000 words in its vocabulary. Therefore, character-based encoding will use fewer tokens compared to word-based encoding.

Issues

This technique helps to shrink the vocabulary size, but it ends up making the sequence longer in character-based encoding. Basically, each word gets split into individual characters, and because of that, the encoded sequence is way longer than the original raw text.

Subword-level Tokenization

The keyword-based encoding algorithms use the following principles:

- Don't break down common words into smaller subwords.
- Split rare words into meaningful subwords.

Byte-Pair Encoding(BPE)

The BPE method will tally the frequency of subwords appearing together and look to merge them if they have the highest occurrence rate. This merging process of subwords continues until there are no more subwords left to combine. Eventually, we end up with a set of subwords that can represent every word in the entire text corpus.

This process includes the following steps:

Step 1: Initialize the vocabulary.

Step 2: Represent each word in the text corpus as a combination of characters with the token <\w> at the end to mark the end of a word (the reason for adding this token will be explained later).

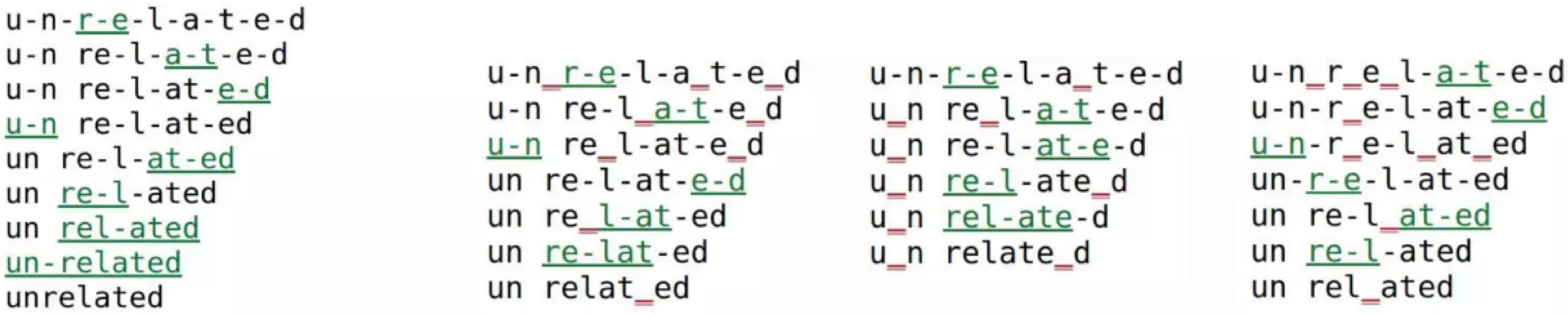
Step 3: Count the frequency of each pair of tokens in the vocabulary.

Step 4: Merge the most frequently occurring pairs to form new character-level n-grams for the vocabulary.

Model	BPE	WordPiece	Unigram
Training	**Starts from a small vocabulary and learns rules to merge tokens**	Starts from a small vocabulary and learns rules to merge tokens	Starts from a large vocabulary and learns rules to remove tokens
Training step	Merges the tokens corresponding to the most common pair	Merges the tokens corresponding to the pair with the best score based on the frequency of the pair, privileging pairs where each individual token is less frequent	Removes all the tokens in the vocabulary that will minimize the loss computed on the whole corpus
Learning	**Merge rules and a vocabulary**	Just a vocabulary	A vocabulary with a score for each token
Encoding	Splits a word into characters and applies the merges learned during training	Finds the longest subword starting from the beginning that is in the vocabulary, then does the same for the rest of the word	Finds the most likely split into tokens, using the scores learned during training

BPE-Dropout

BPE-dropout - simple and effective subword regularization method based on and compatible with conventional BPE that stochastically corrupts the segmentation procedure of BPE, which leads to producing multiple segmentations within the same fixed BPE framework



BPE (a) BPE-Dropout (b)

xbert thường : loss: 0.2514, Accuracy: 0.8392

dropout = 0.2: loss: 0.2570, Accuracy: 0.8516

dropout = 0.1: loss: 0.2544, Accuracy: 0.8464

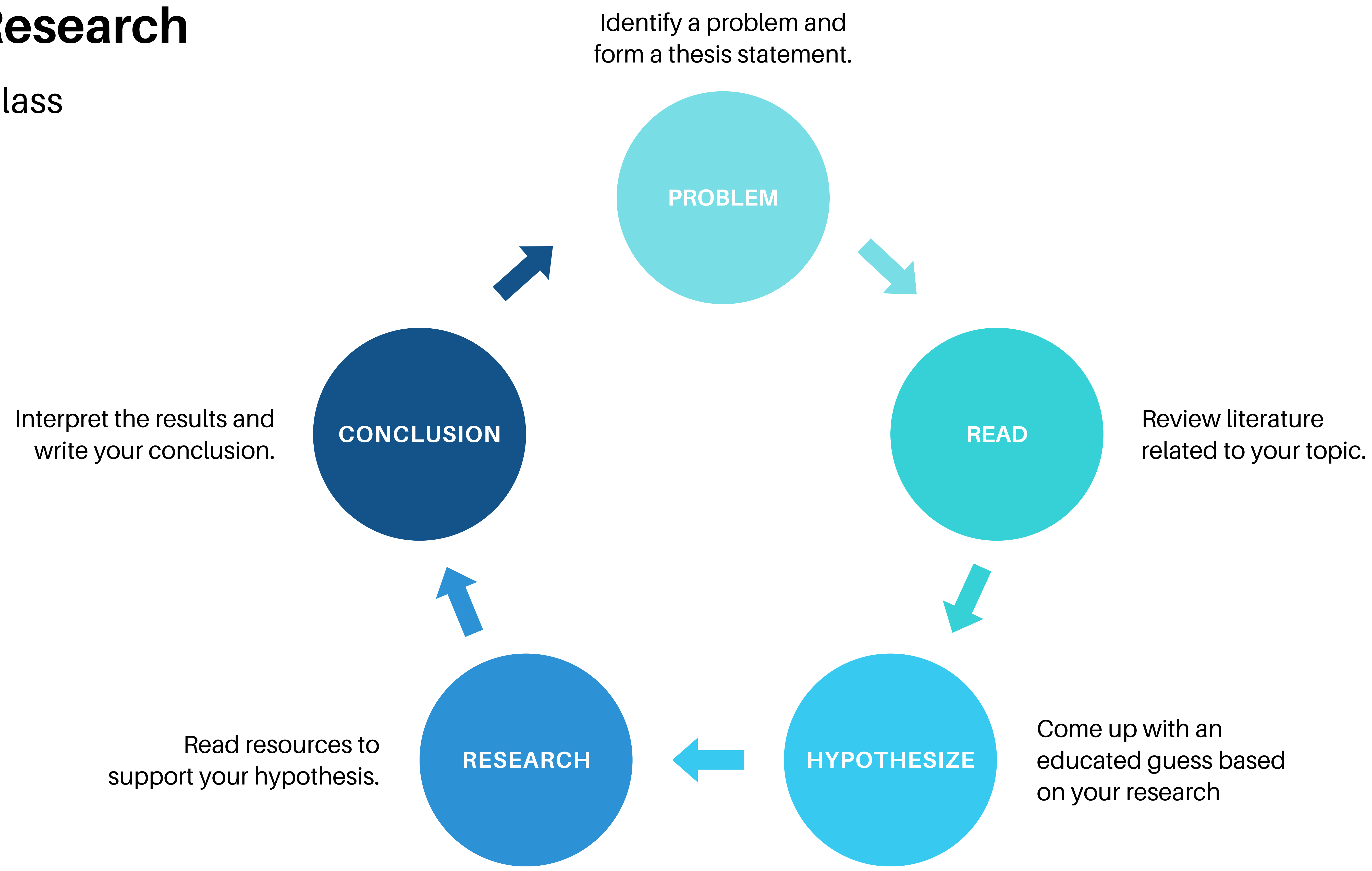
dropout = 0.4: loss: 0.2613, Accuracy: 0.8671

dropout = 0.3: loss: 0.2618, Accuracy: 0.8614

dropout = 0.5: loss: 0.2616, Accuracy: 0.8695

The Stages of Research

Miss Jones Science Class



The Stages of Research

Miss Jones Science Class

1. **PROBLEM**

   Identify a problem and form a thesis statement.

2. **READ**

   Review literature related to your topic.

3. **HYPOTHESIZE**

   Come up with an educated guess based on your research.

4. **RESEARCH**

   Read resources to support your hypothesis.

5. **CONCLUSION**

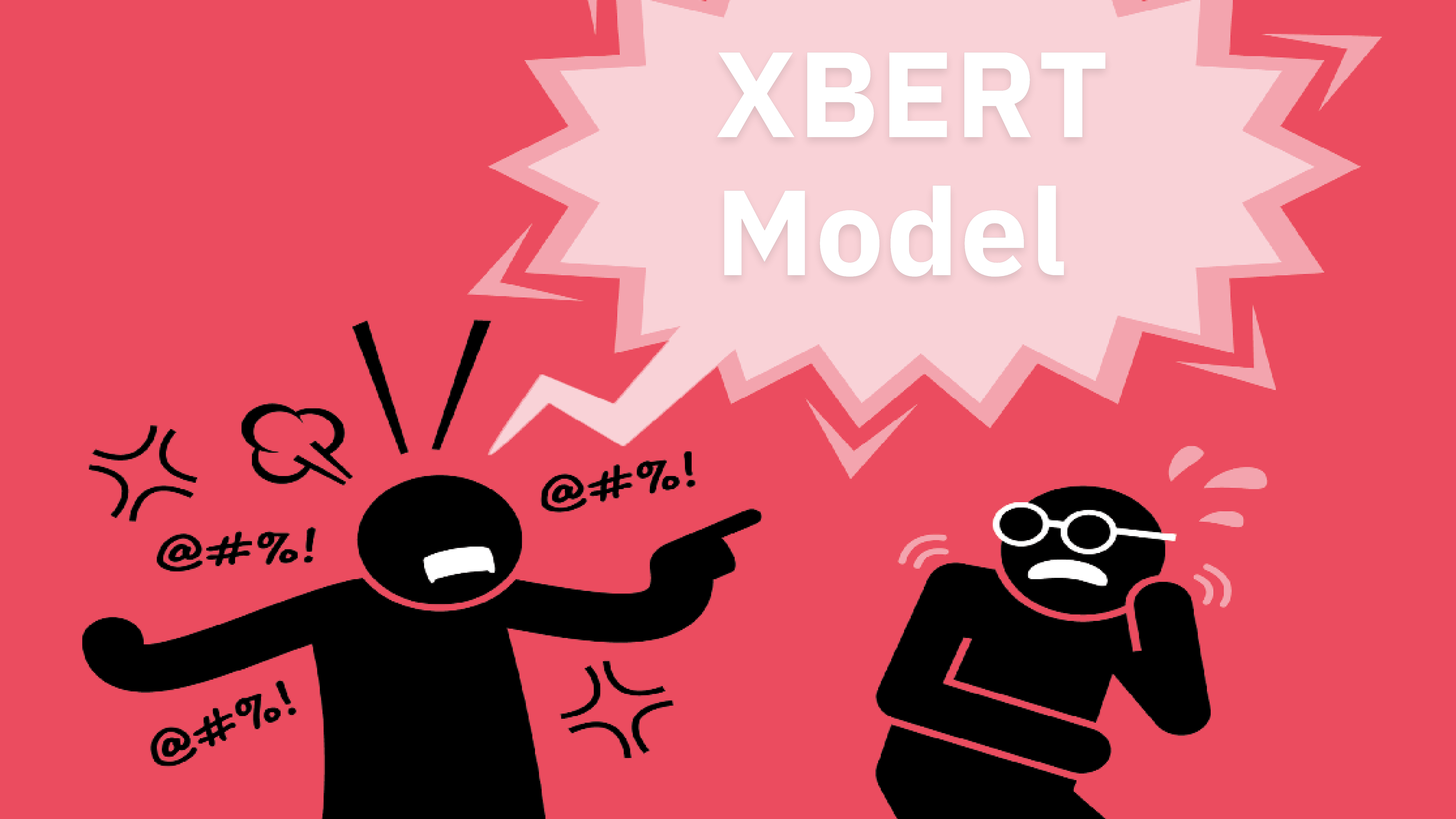
   Interpret the results and write your conclusion.

The Stages of Research

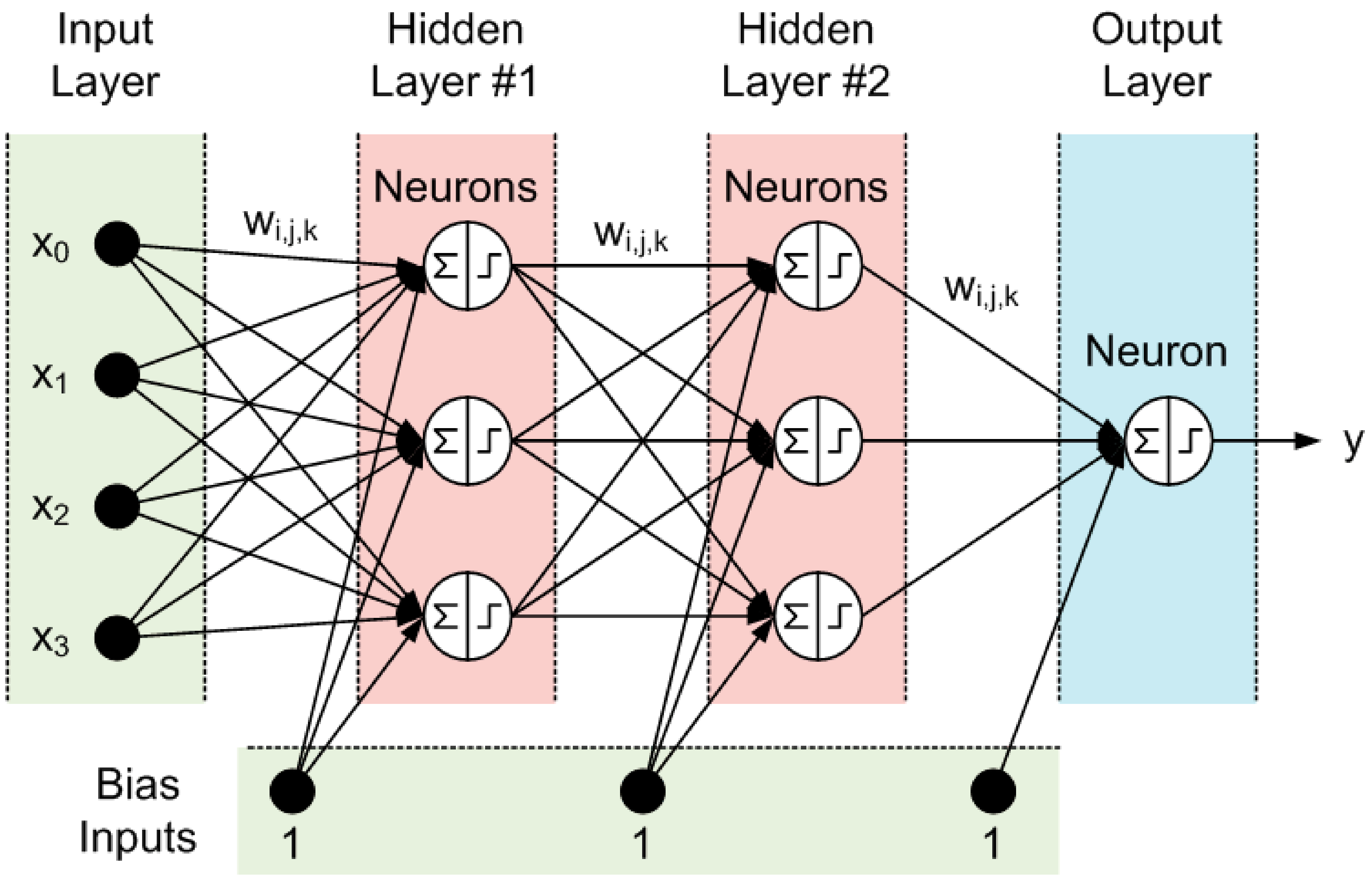
Miss Jones Science Class

PROBLEM	READ	HYPOTHESIZE	RESEARCH	CONCLUSION

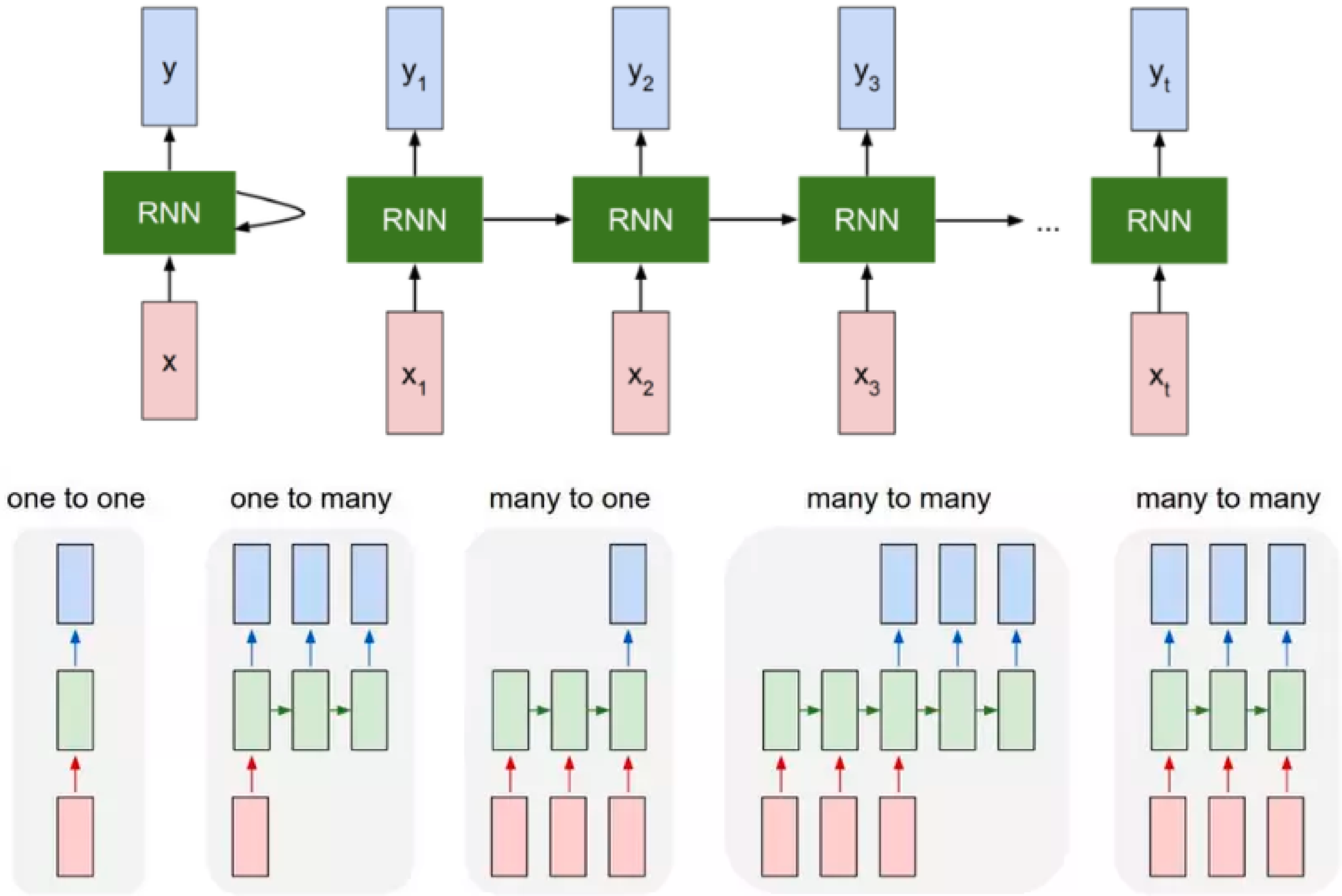
Highlight two or more cells, **right-click** then choose
"Merge Cells" to organize your table according to your needs!


XBERT
Model
@#%!
@#%!
@#%!

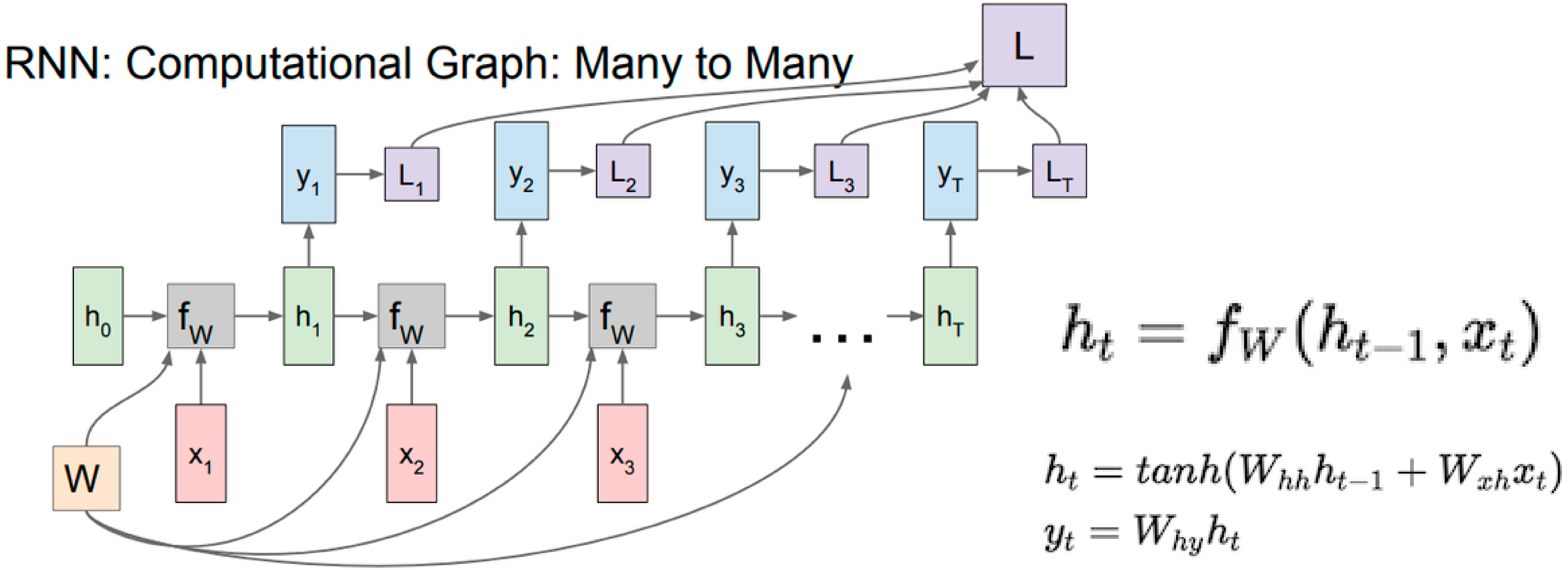
Neural Network



RNNs

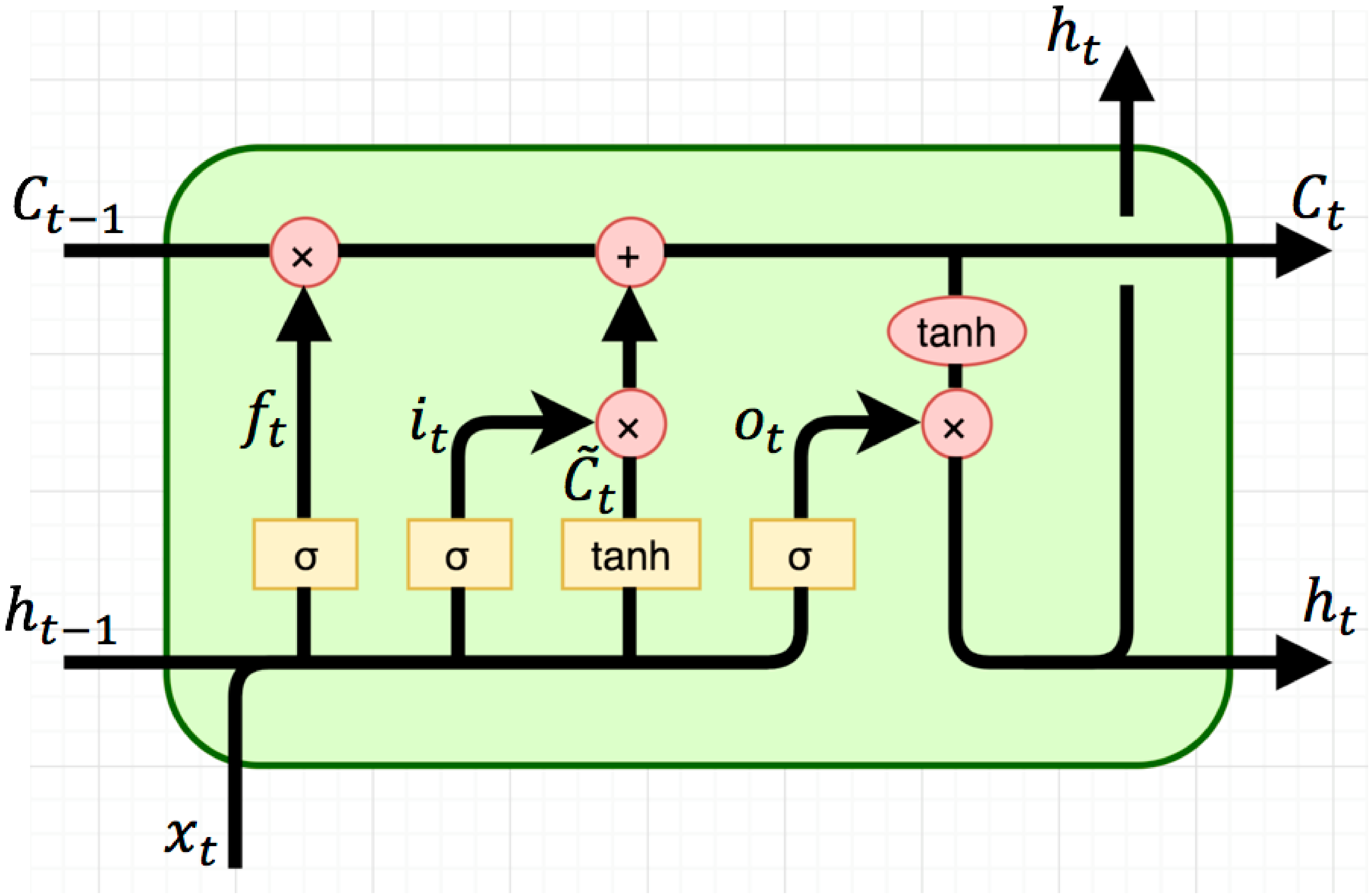

y
RNN
x
y_1
y_2
y_3
y_t
RNN
RNN
RNN
...
RNN
x_1
x_2
x_3
x_t
one to one
one to many
many to one
many to many
many to many

RNNs



RNNs (https://viblo.asia/p/recurrent-neural-networkphan-1-tong-quan-va-ung-dung-jvElaB4m5kw#_mang-hoi-quy-rnn-0)

Long-Short Term Memory



Long-Short Term Memory

- Forget gate: $f_t = \sigma(U_f * x_t + W_f * h_{t-1} + b_f)$
- Input gate: $i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i)$
- Output gate: $o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o)$

σ : sigmoid

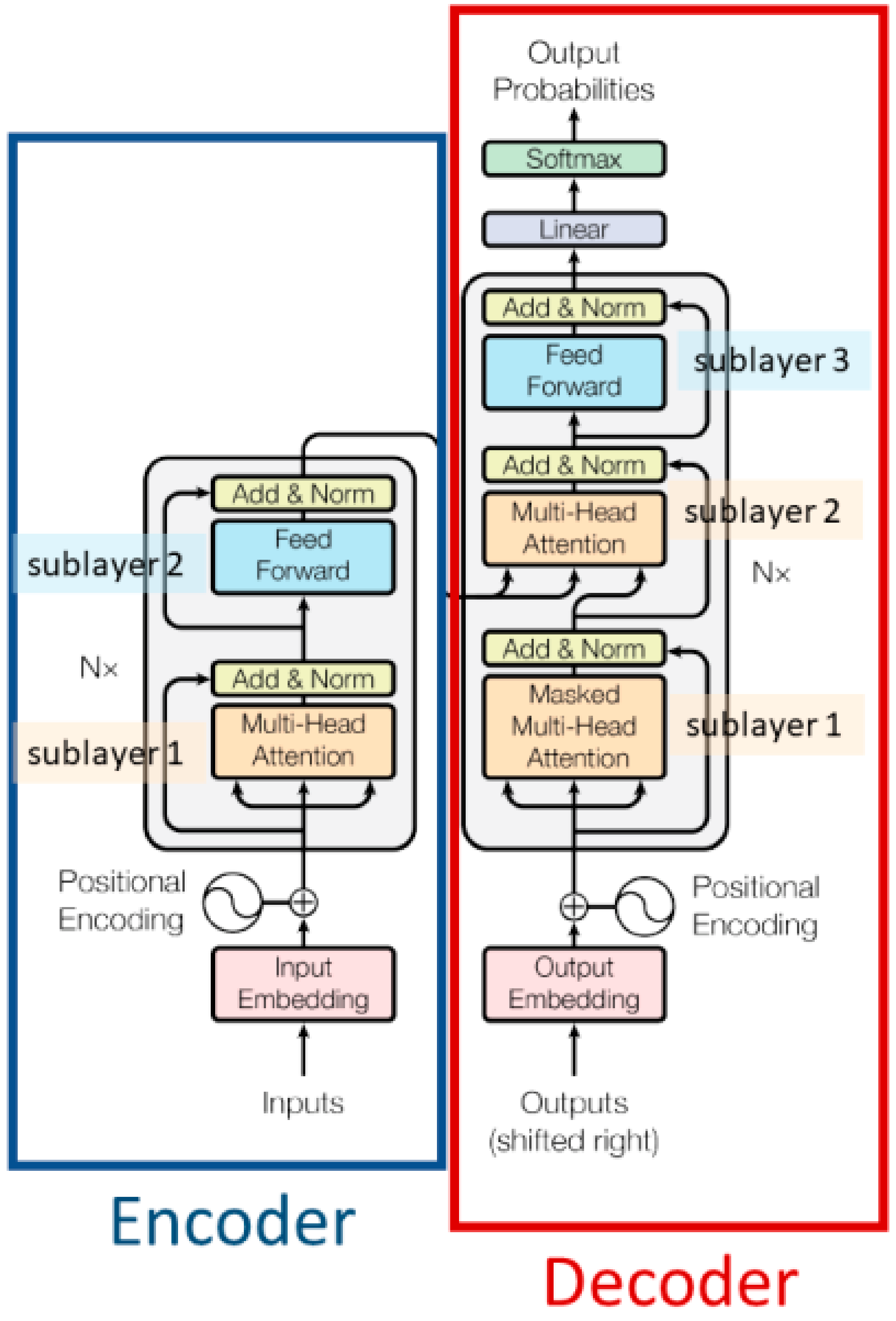
U_f, U_i, U_o : Current input link weights x_t

W_f, W_i, W_o: Previous hidden state link weights with h_t-1

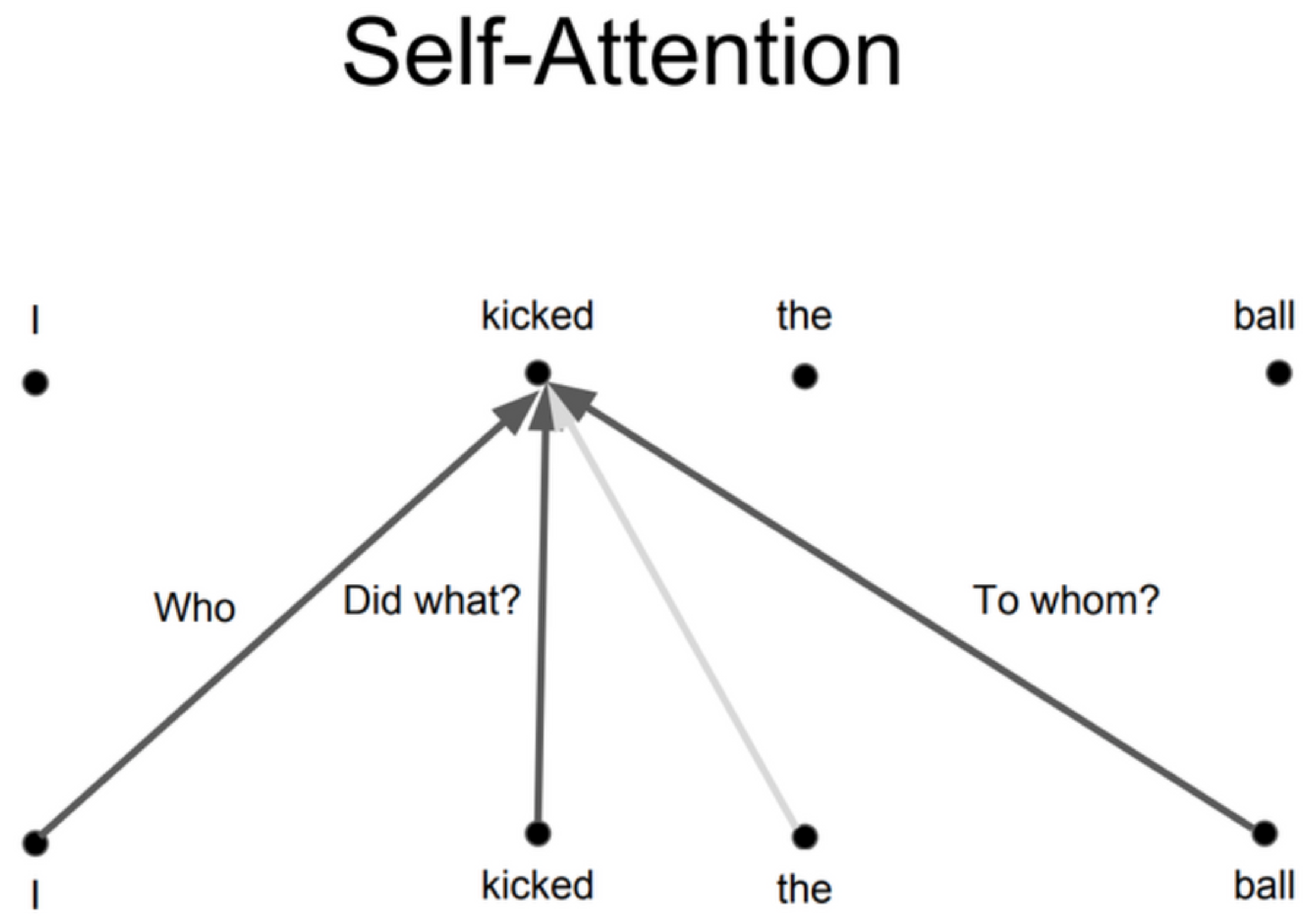
b_f, b_i, b_o: Bias for gate

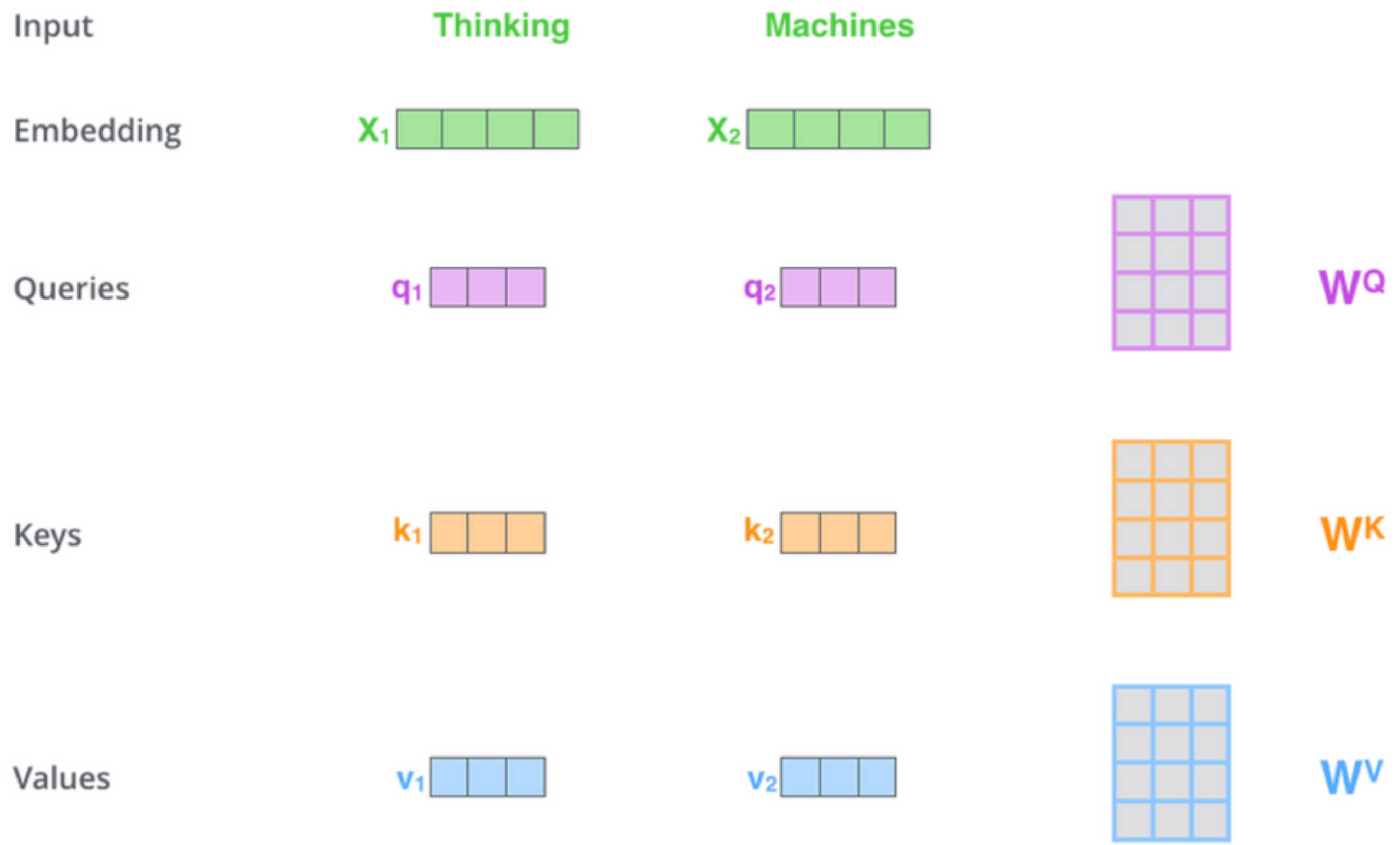
Transformer :

Attention all you need



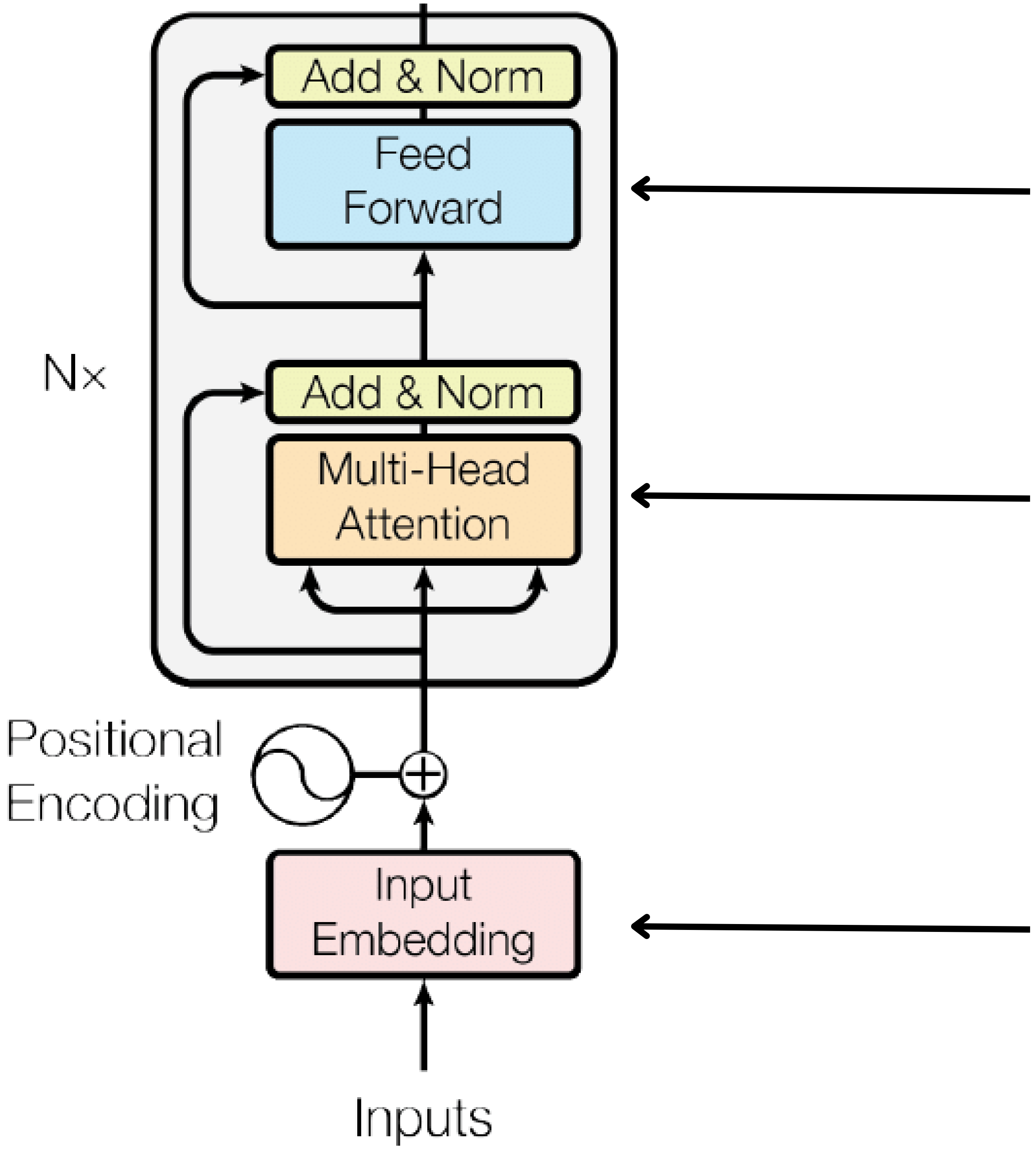
Self-Attention





$$Z = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{Dimension\ of\ vector\ Q, K \text{ or } V}}\right) \cdot V$$

Transformer



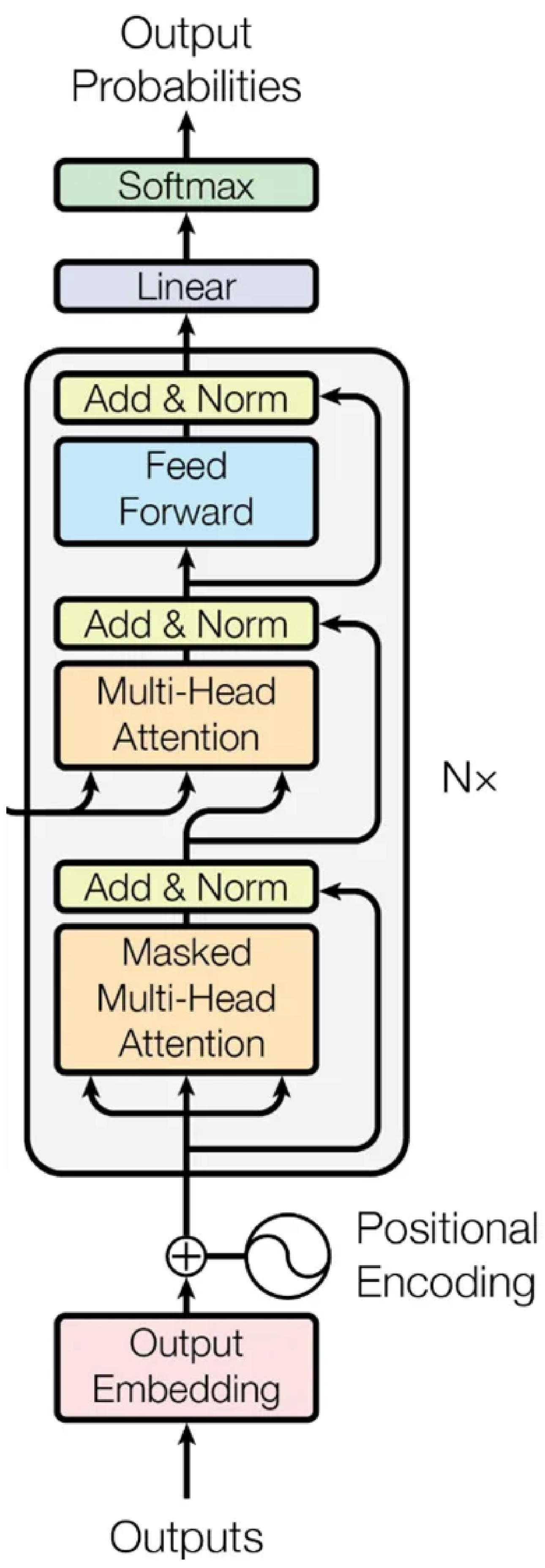
Feed Forward

Multi-Head Attention

Input Embedding :

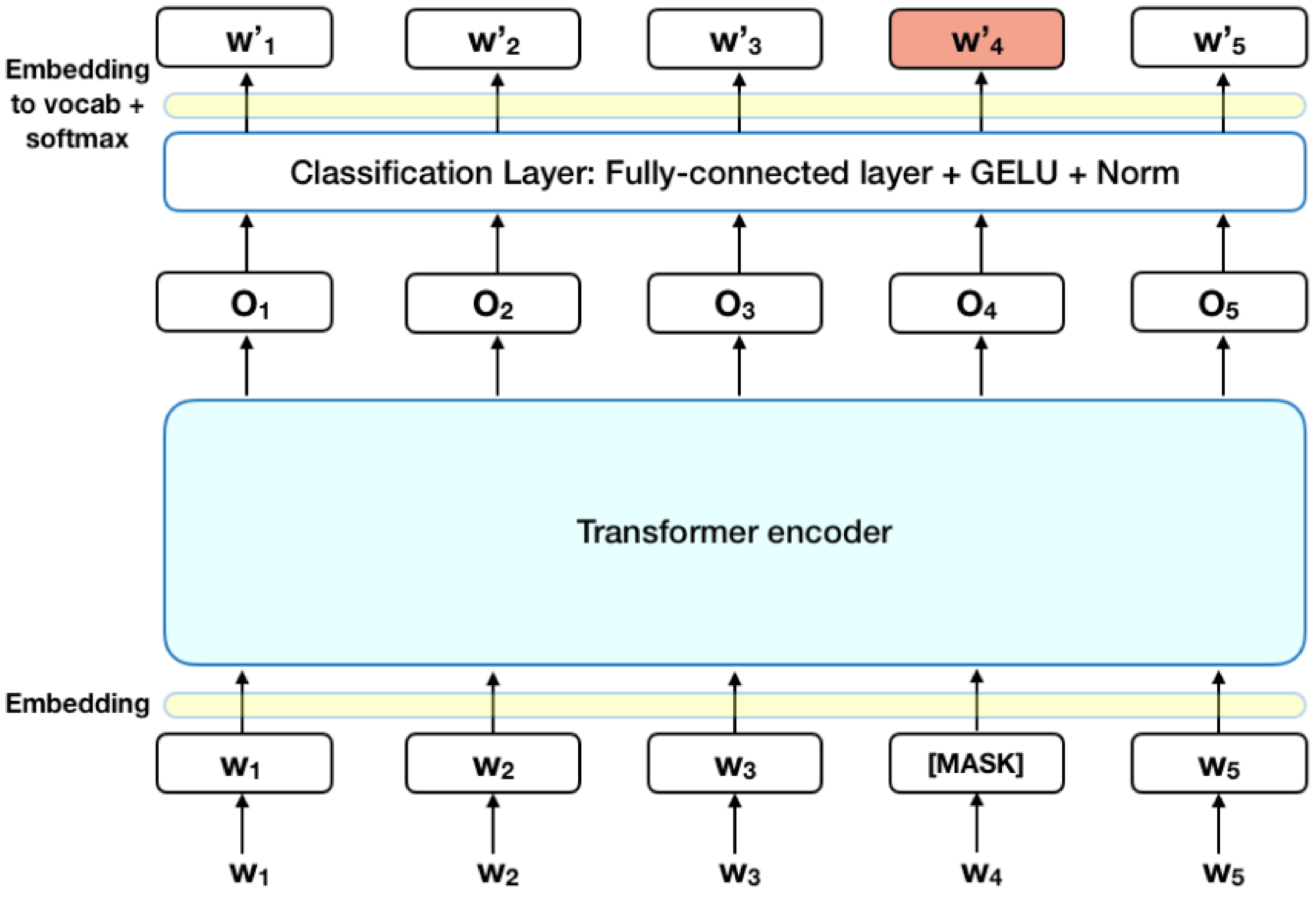
Glove, Fasttext, Word2Vec, ..

Transformer



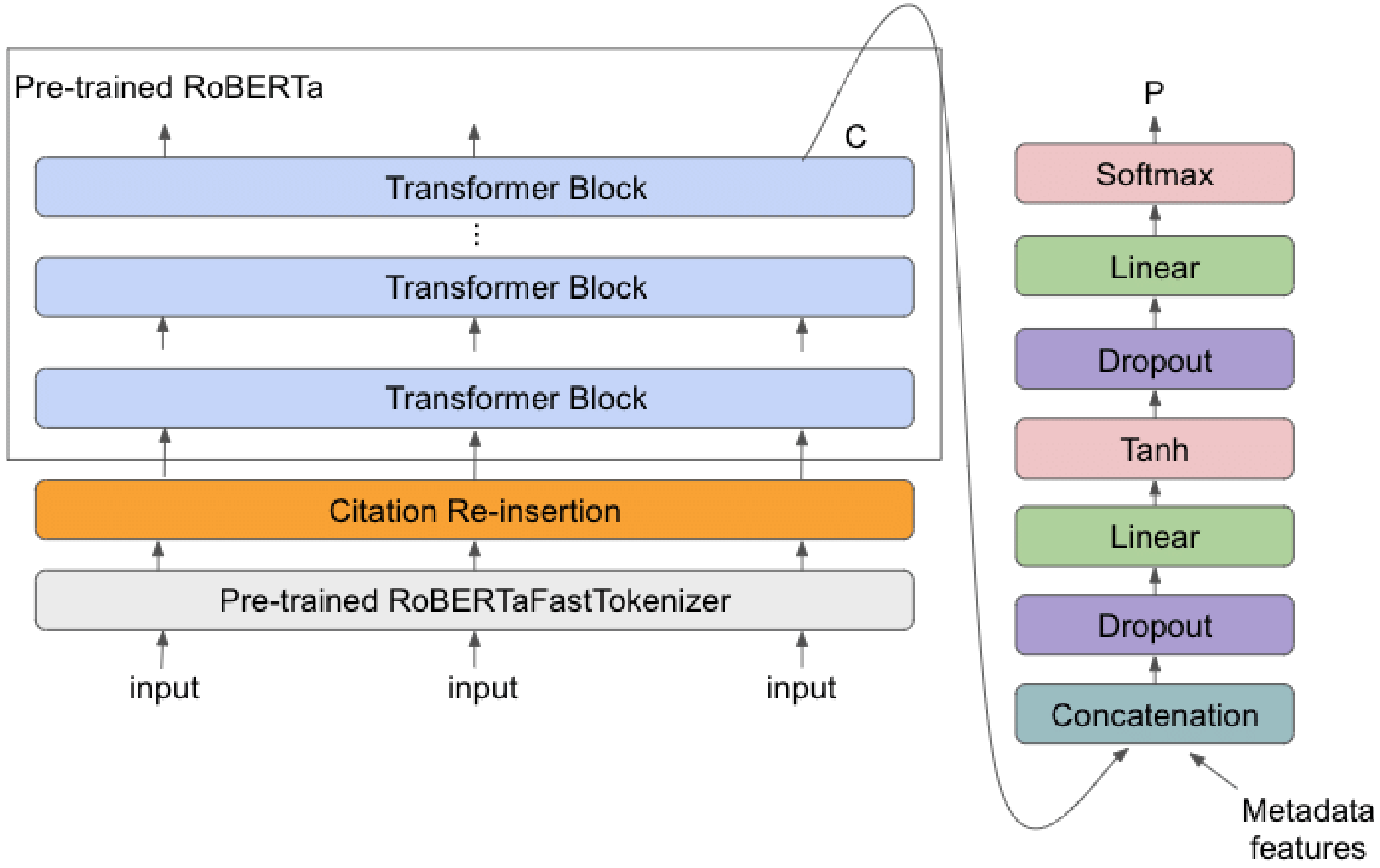
Masked Multi-Head Attention

BERT



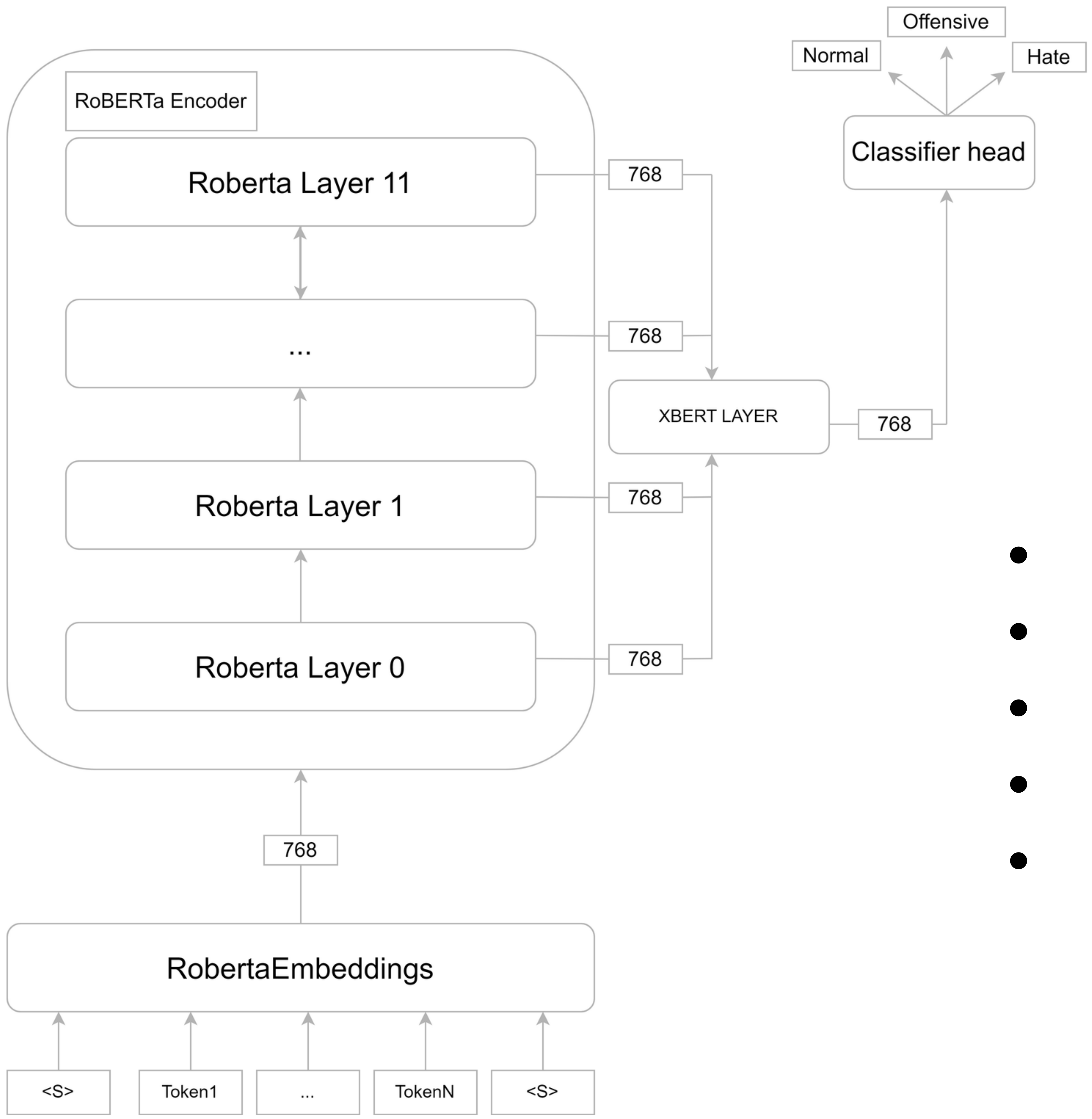
w'1
w'2
w'3
w'4
w'5
Embedding to vocab + softmax
Classification Layer: Fully-connected layer + GELU + Norm
O1
O2
O3
O4
O5
Transformer encoder
Embedding
w1
w2
w3
[MASK]
w5
w1
w2
w3
w4
w5

RoBERTa



- Pretraining is Longer and with Bigger Data
- Skip Next Sentence Prediction (NSP) Tasks
- Use Dynamic Masking
- Optimization Hyperparameters

XBERT




```
(Xbert): Sequential(
  (0): Linear(in_features=768, out_features=768, bias=True)
  (1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  (2): Dropout(p=0.1, inplace=False)
)
```

- K-fold = 10 with 10 epochs
- EDA
- max_length = 60, batch_size =128
- StepLR(optimizer, step_size=70, gamma=0.9)
- X-BERT tokenizer

Training model

Trying a lot of method to training have a best model by :

- Compare 3 models with **num_attention_head** = 12,16,**32**
- Compare 3 models with **batch_size** = 32, 64, **128**
- Compare model **Roberta** vs **XBert** on VLSP with autoLR
- Combine 2 dataset **VLSP**, **ViHSD**
- Training 100 epochs on each dataset
- Compare model with **Xbert tokenization** and **PhoBert tokenization**
- Compare 2 methods **KFold=10** and **Normal**
- Combine 3 datasets to have the best model : **VLSP**, **ViHSD**, **social_media**(from crawling social_network)
- Training 5 models Xbert(with **Xbert_Tokenize dropout from 0.1-0.5**) to choose the model have best acc => choose : **dropout=0.3**
- Label dataset **test_VLSP** by model **Xbert (VLSP+ViHSD+social)** and send mail to VLSP Resources

EVALUATION

F1-SCORE

F1-score: performance evaluation for classification.

$$F_1 = \frac{2}{Recall^{-1} + Precision^{-1}}$$

F1-MACRO

F1-macro: computed as mean of F1 scores for each class.

$$F_{1-Macro} = \frac{F_{1-HATE} + F_{1-OFFENSIVE} + F_{1-CLEAN}}{3}$$

Result of Training

Model	VLSP		ViHSD	
	Accuracy	Macro-F1	Accuracy	Macro-F1
PhoBert	94.1	66.03	86.61	53.0
PhoBert-CNN	98.26	90.89	87.17	64.43
XBert	**99.75**	**98.05**	**96.55**	**91.67**

PERFORMANCE

Training time

```
###### Epoch 1/10 ######
Epoch 1/10:  25%|██        | 43/170 [00:20<00:53,  2.39it/s]
```

PhoBert

```
###### Epoch 1/10 ######
Epoch 1/10:  10%|█         | 17/170 [00:23<03:12,  1.26s/it]
```

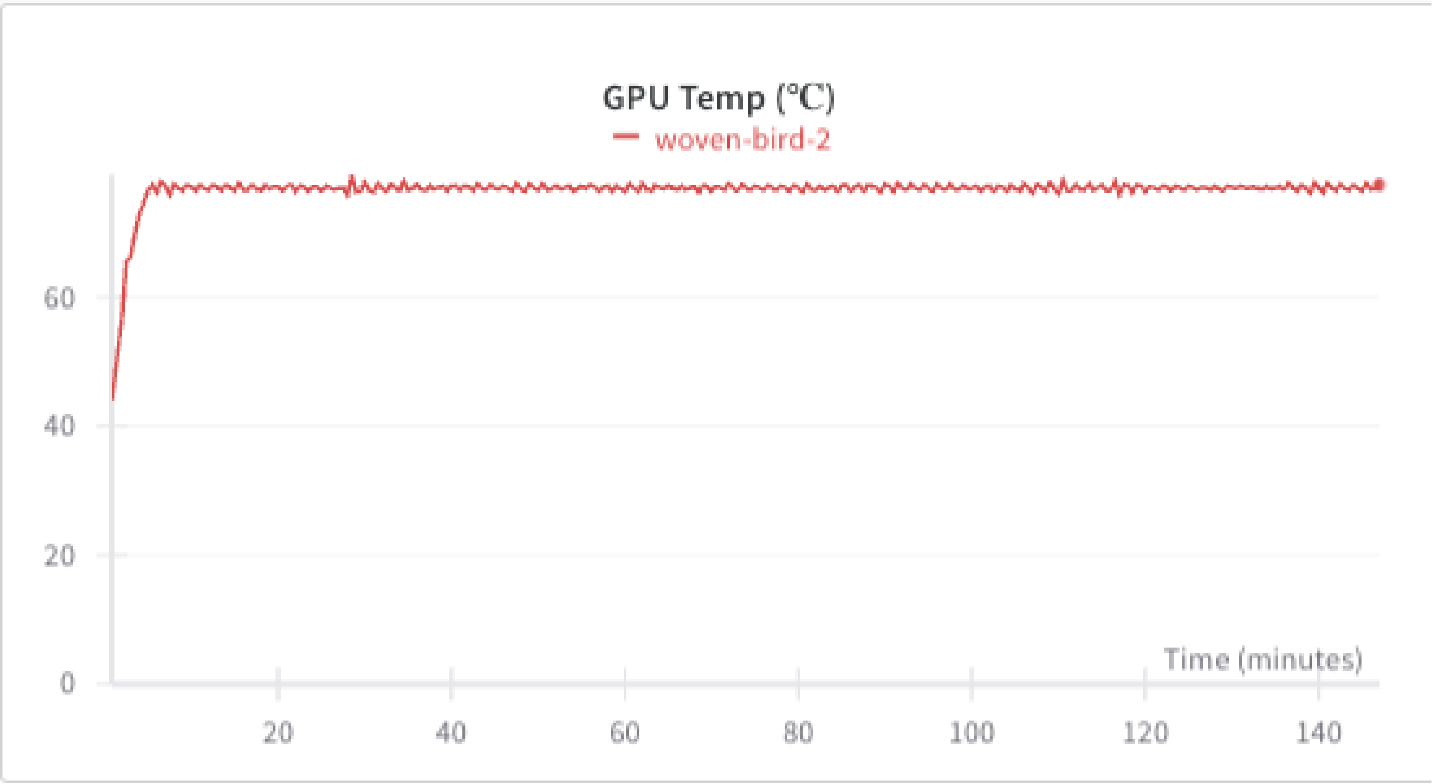
XBert

```
###### Epoch 1/10 ######
Epoch 1/10:  16%|██        | 28/170 [00:13<01:06,  2.15it/s]
```

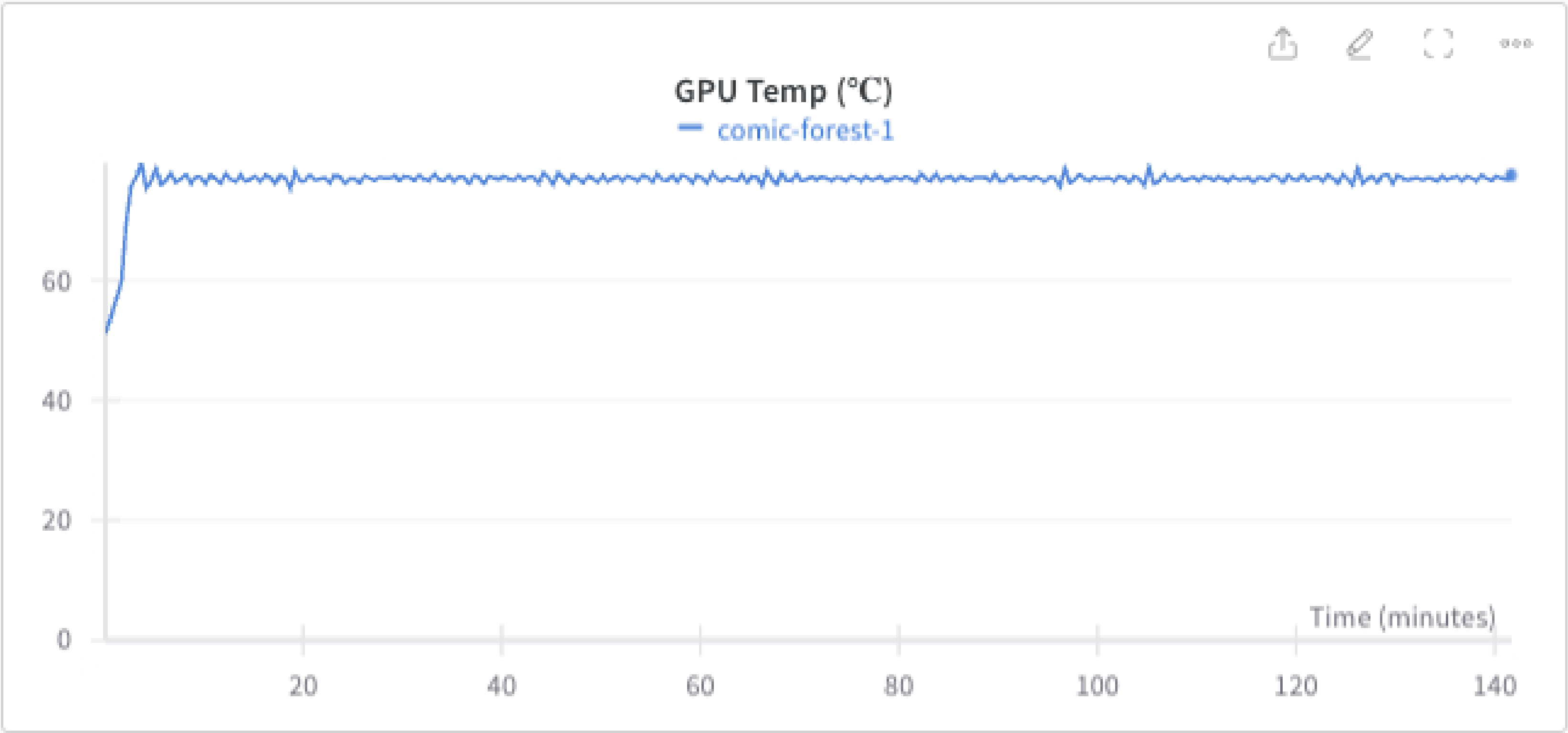
PhoBert-CNN

PERFORMANCE

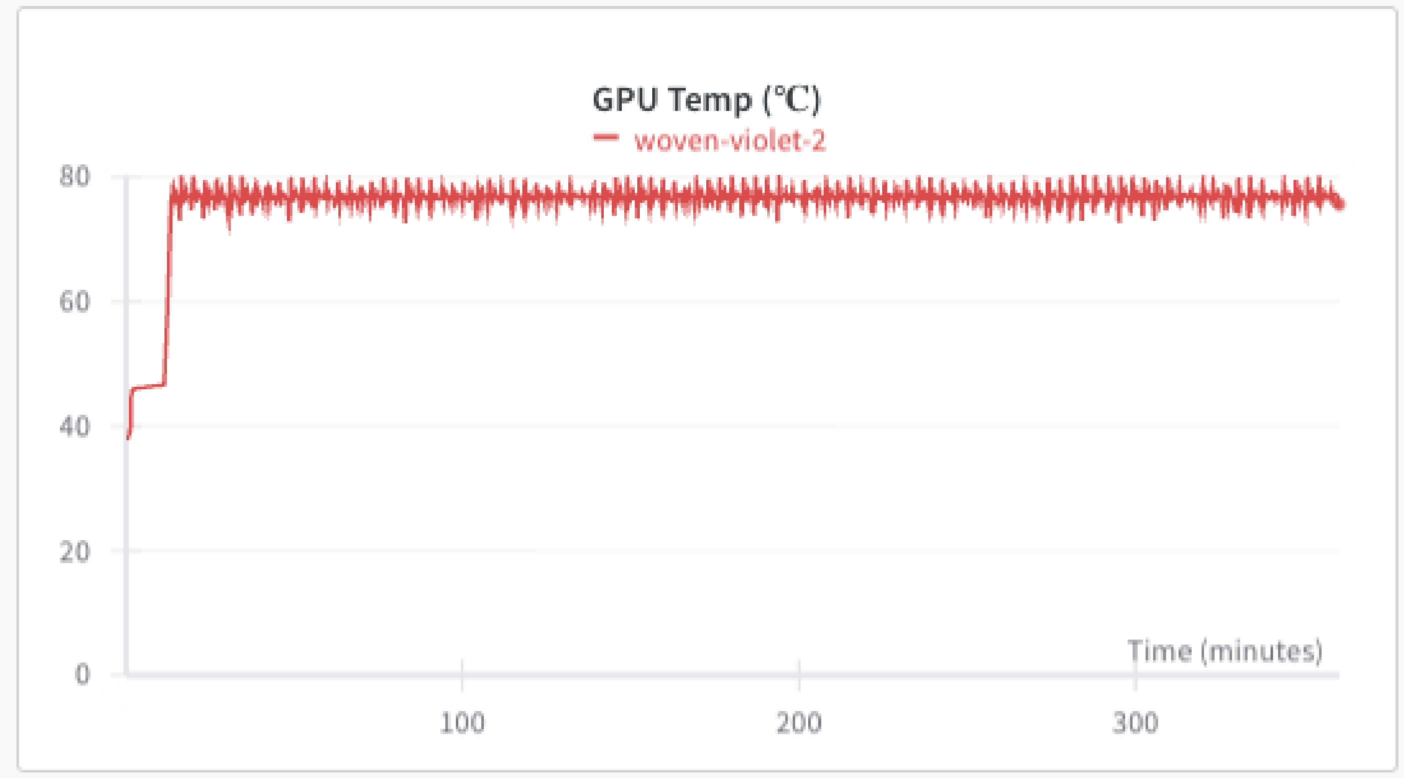
GPU Temp



PhoBERT



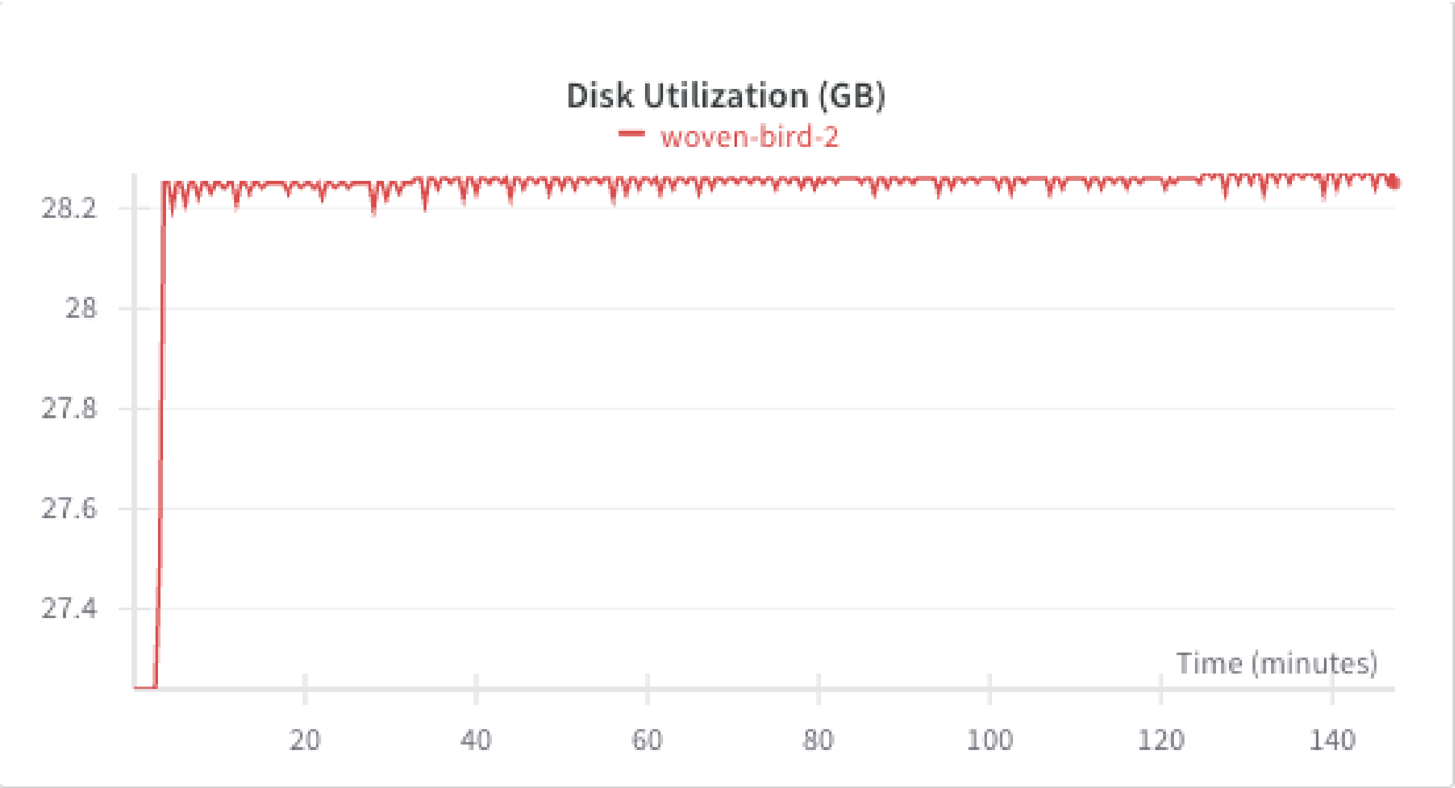
PhoBERT - CNN



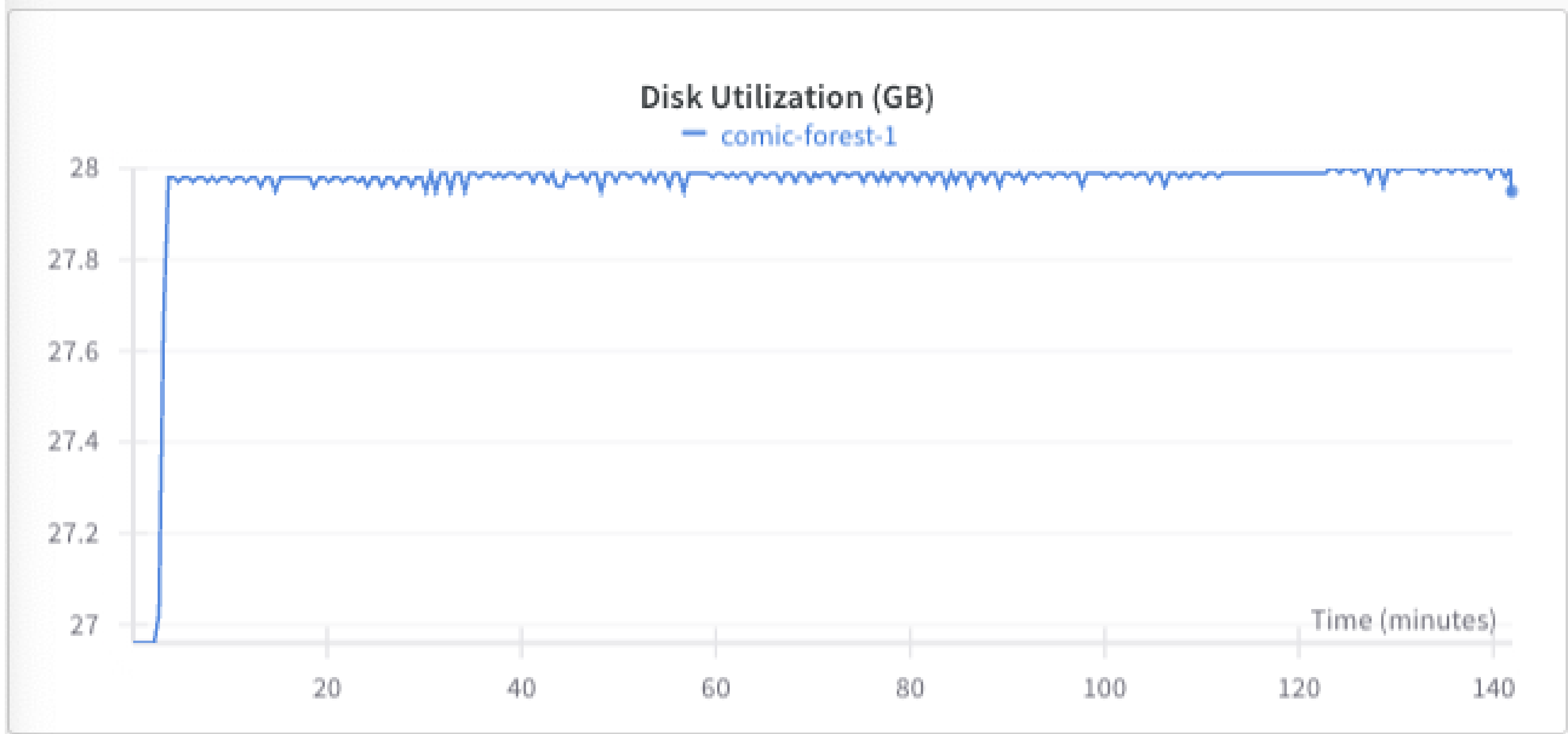
XBERT

PERFORMANCE

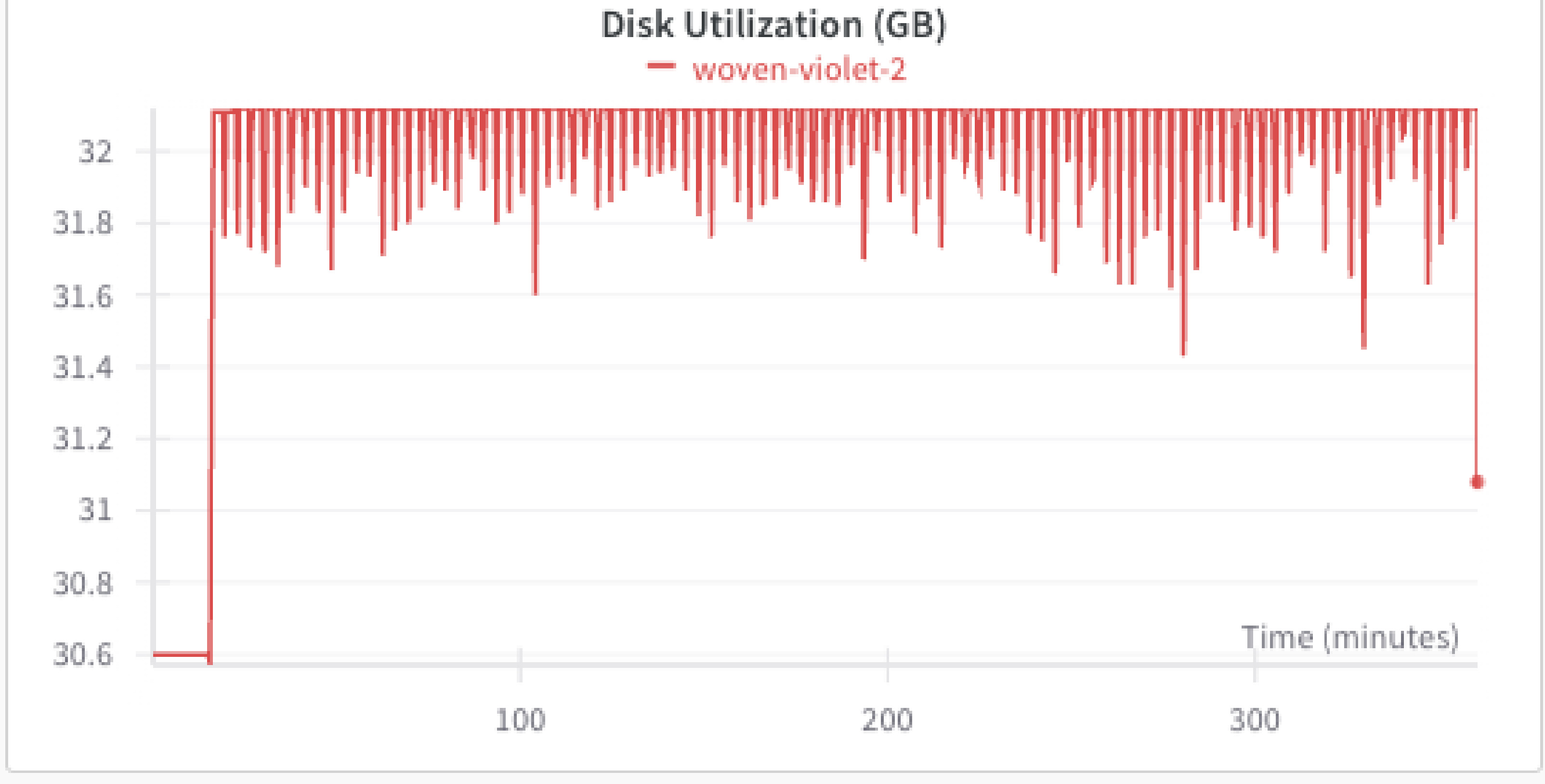
GPU Temp



PhoBERT



PhoBERT - CNN



XBERT

THANK YOU FOR LISTENING