KiSTi 한국과학기술정보연구원 Korea Institute of Science and Technology Information

2022

Advanced Topic in Research Data-centric Deep Learning

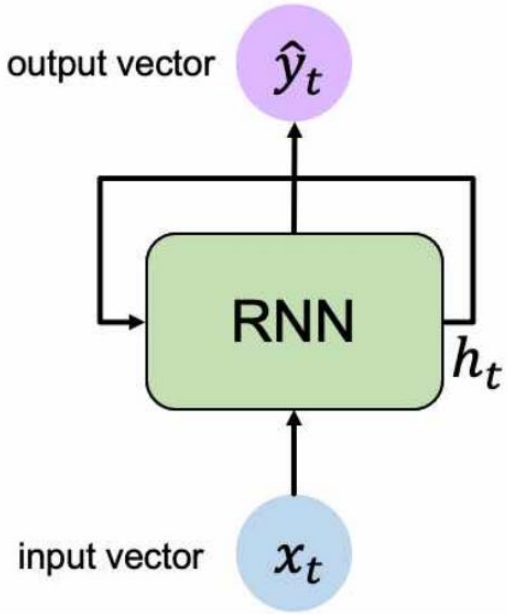
Lec 10: Seq2Seq and Attention

hsyi@kisti.re.kr

Hongsuk Yi (이홍석)

Reviewing the last class: RNN and LSTM

RNN State Update and Output



Output Vector

$$\hat{y}_t = \boldsymbol{W}_{hy}^{T} h_t$$

Update Hidden State

$$h_t = \tanh(\boldsymbol{W}_{hh}^{T} h_{t-1} + \boldsymbol{W}_{xh}^{T} x_t)$$

Input Vector

$$x_t$$

RNN example

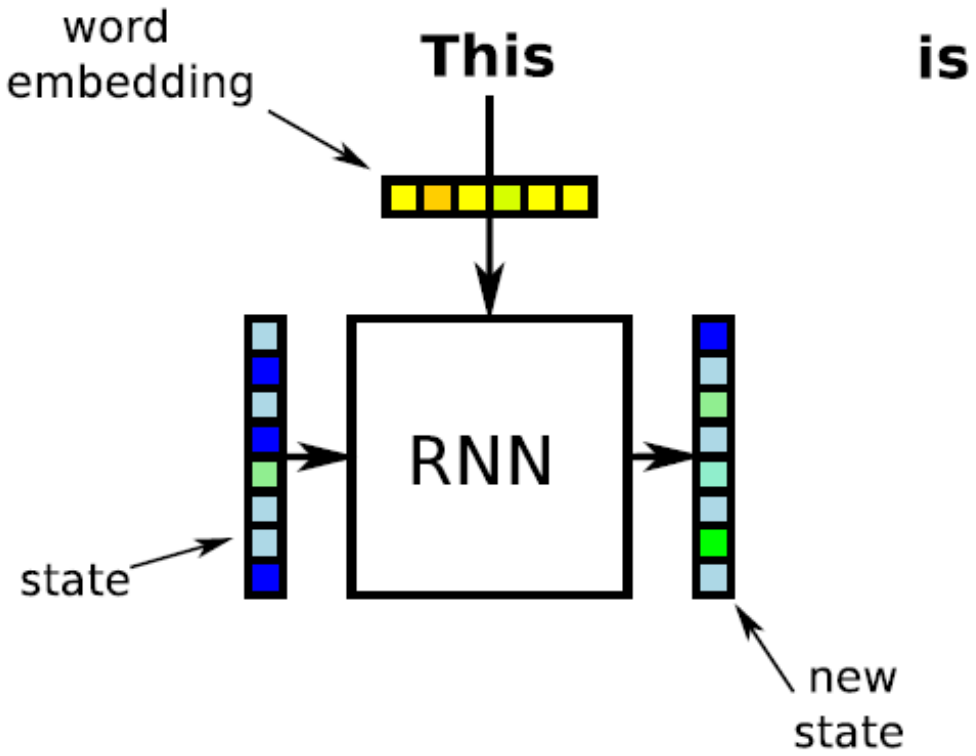


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

example: using the RNN output in a document classifier

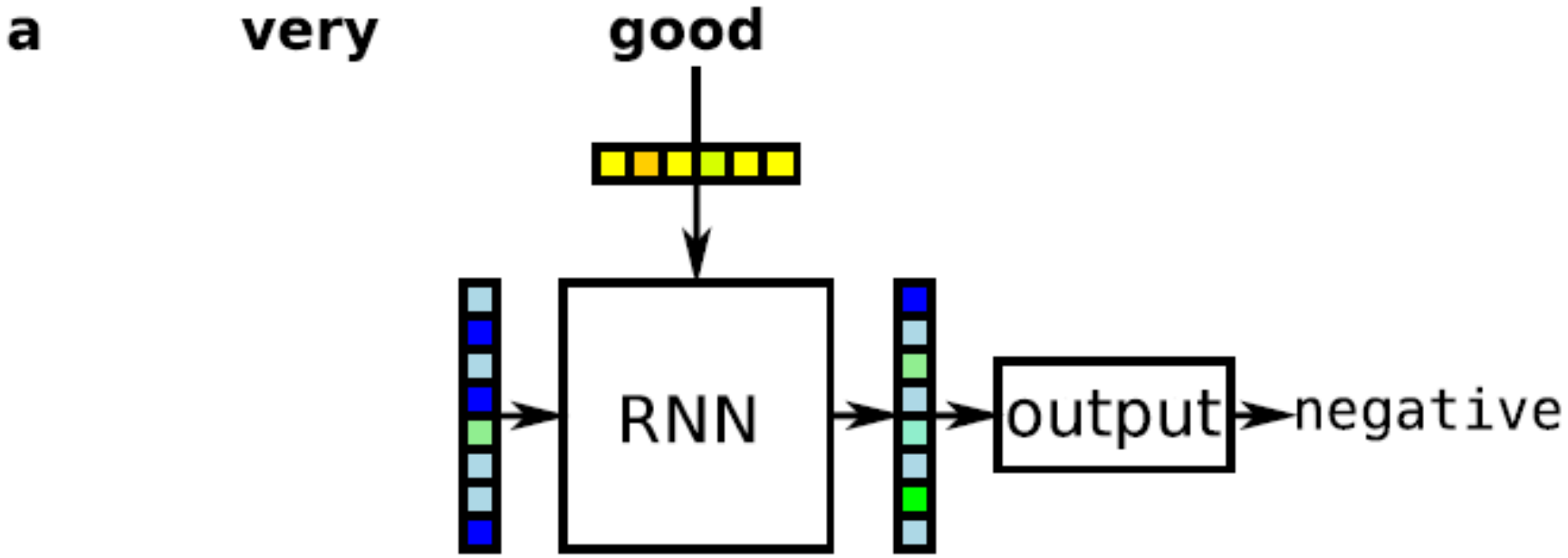


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

simple RNN implementation

- the Elman RNN or simple RNN looks similar to a feedforward NN
  - ✓ the next state is computed like a hidden layer in a feedforward NN
  - ✓ the output is identical to the state representation:

$$y_t = s_t$$

$$s_t = g(\boldsymbol{W} \cdot (s_{t-1} \oplus x_t) + \boldsymbol{b})$$

actication is typically tanh

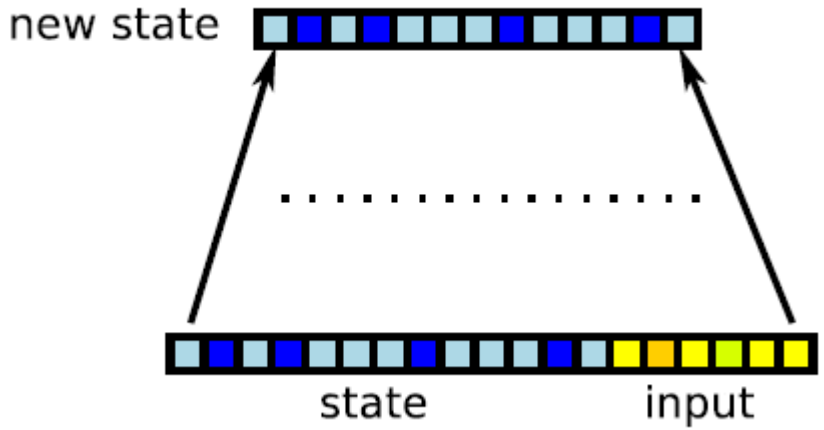


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Gating of LSTM Mechanism

- gating architectures allow information flow to be controlled more carefully
  - ✓ should we copy the previous state, or replace it?
  - ✓ the "gates" are controlled by their own parameters

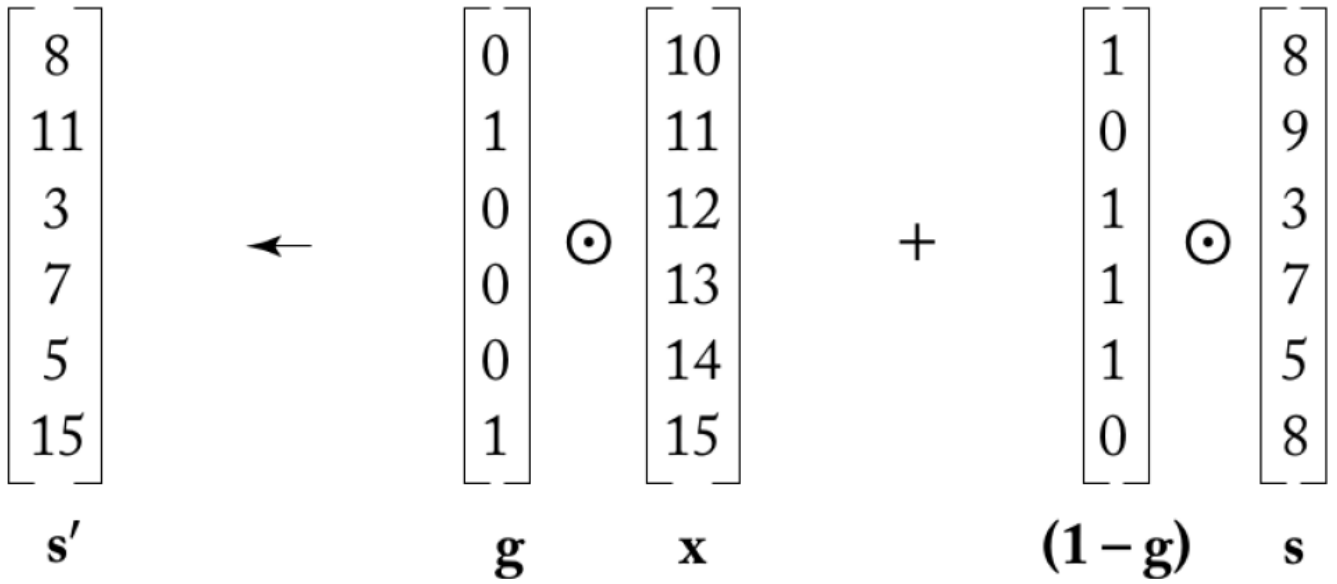


image from Goldberg's book

image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

limitations of RNNs

- even with gated RNNs, it can be hard to cram the useful information into the last state

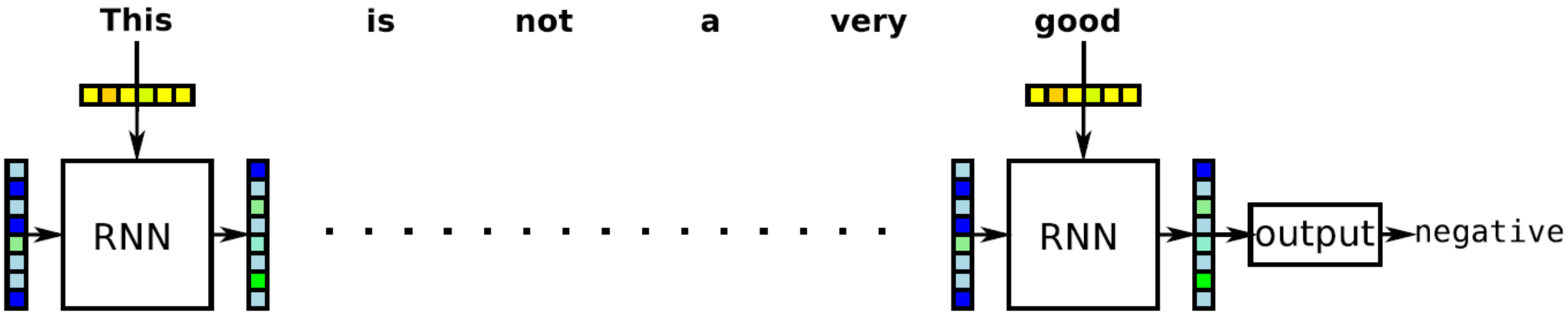


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Today:
Attention Intuition

Attention models: use in NLP applications

- Attention models were first proposed by Bahdanau et al. (2015) in the context of machine translation
  - ✓ today, used in many different applications (Galassi et al., 2019)

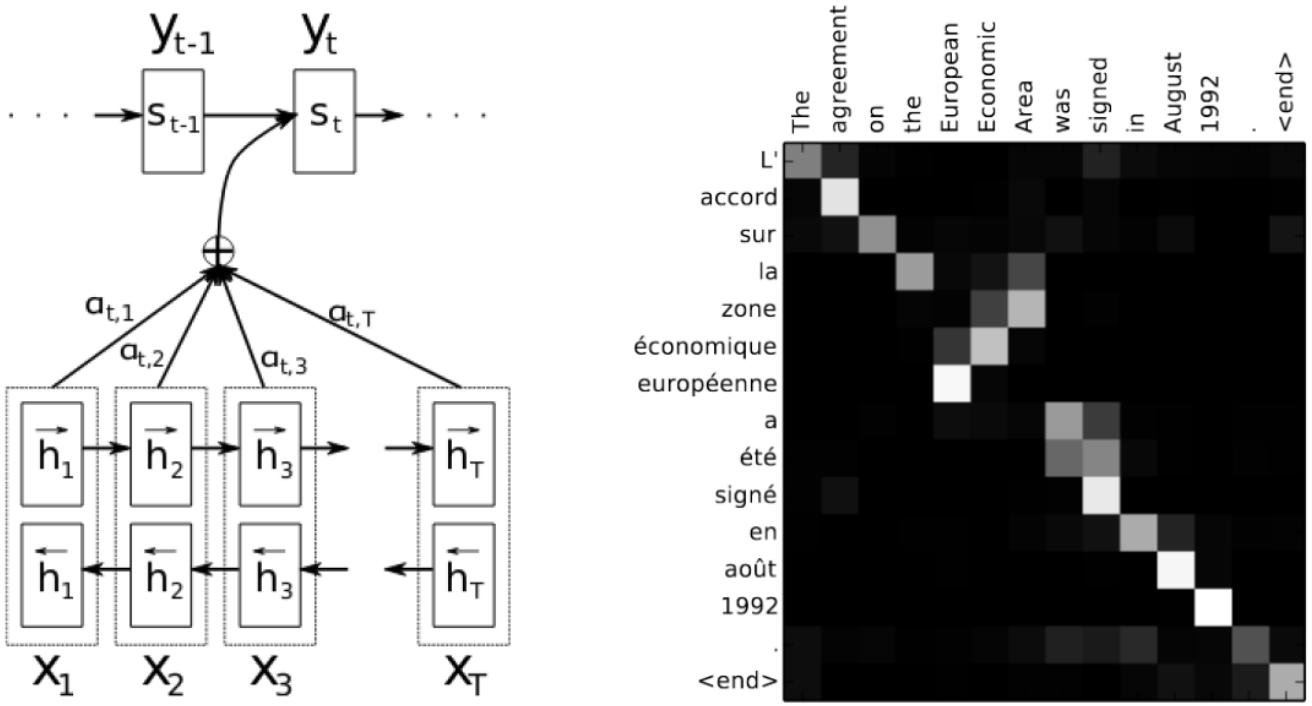


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

attention models: intuition

- In attention models, we consider all the RNN states observed when processing a sequence
  - ✓ we compute a "summary" (weighted average) of the states
    - the weights correspond to some notion of "importance"

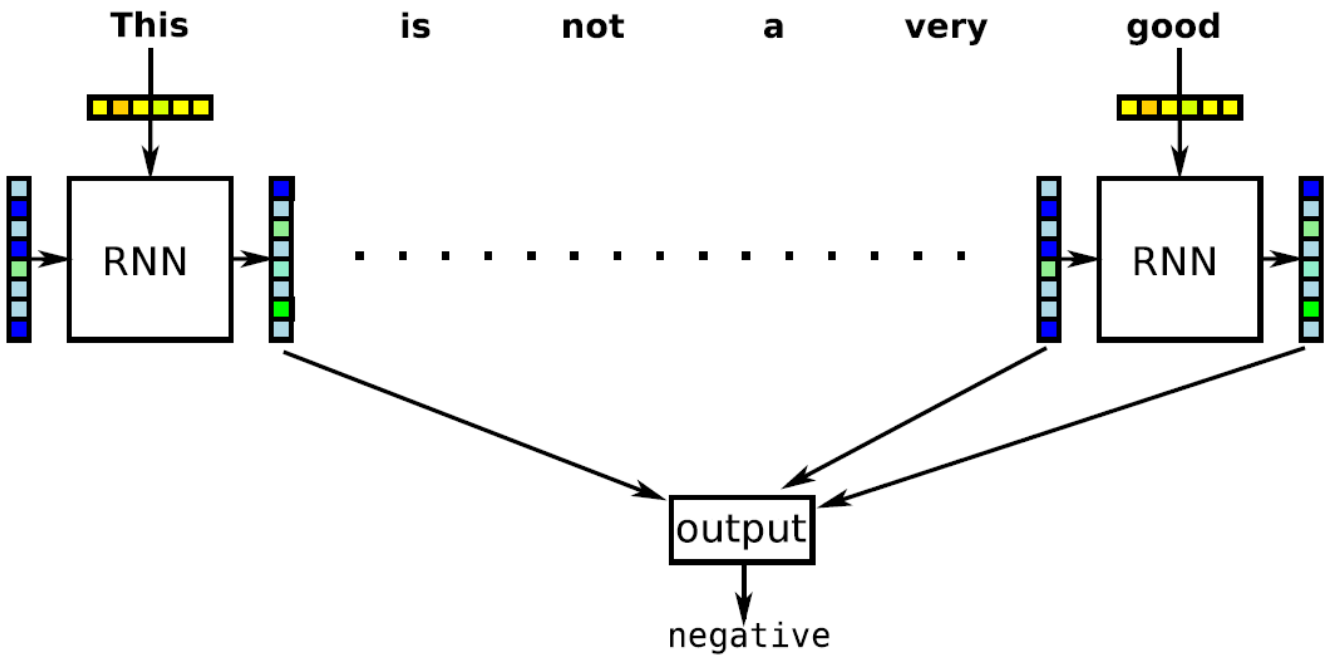


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Attention: a general formulation

- What is “importance score” for each state (h)

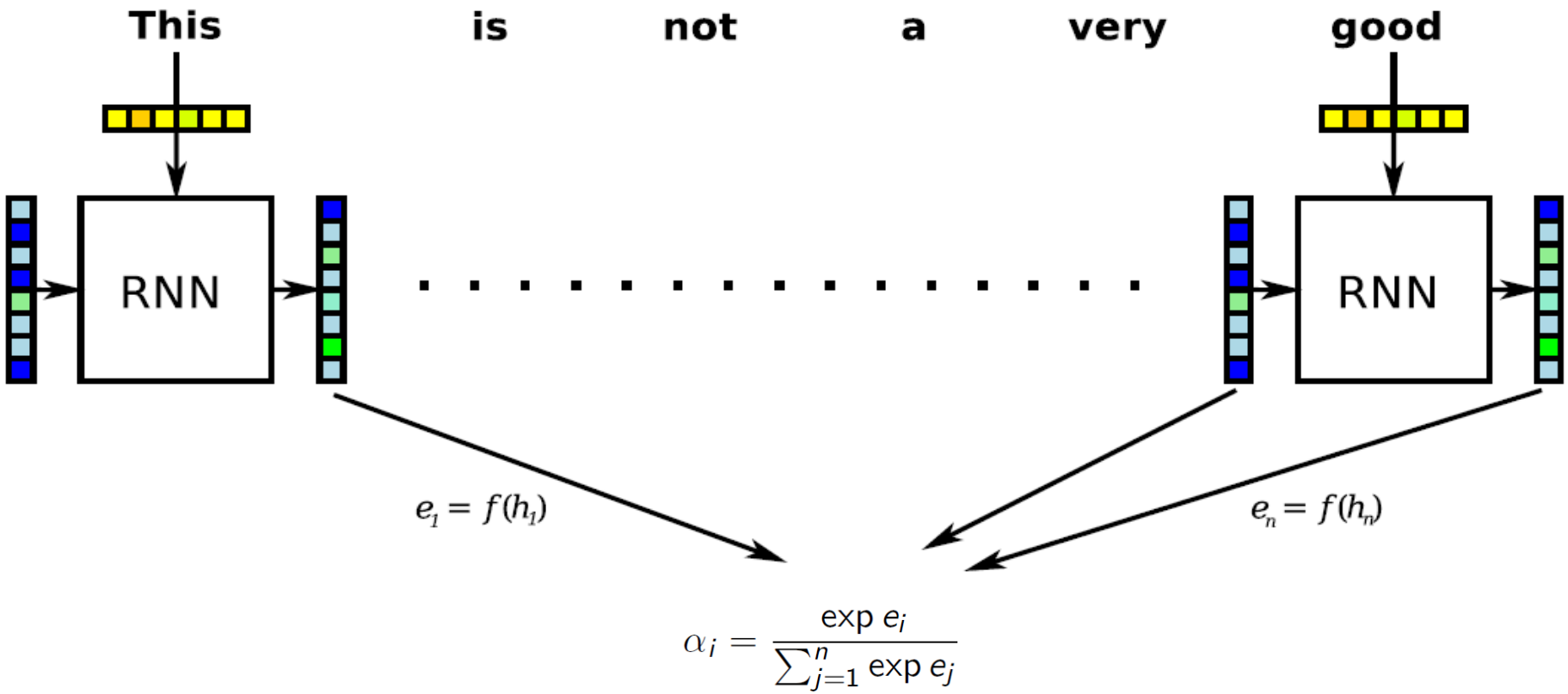


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Attention: a general formulation

- Attention weights, we apply the softmax of the “importance score”
- The “summary” is computed as a weighted sum

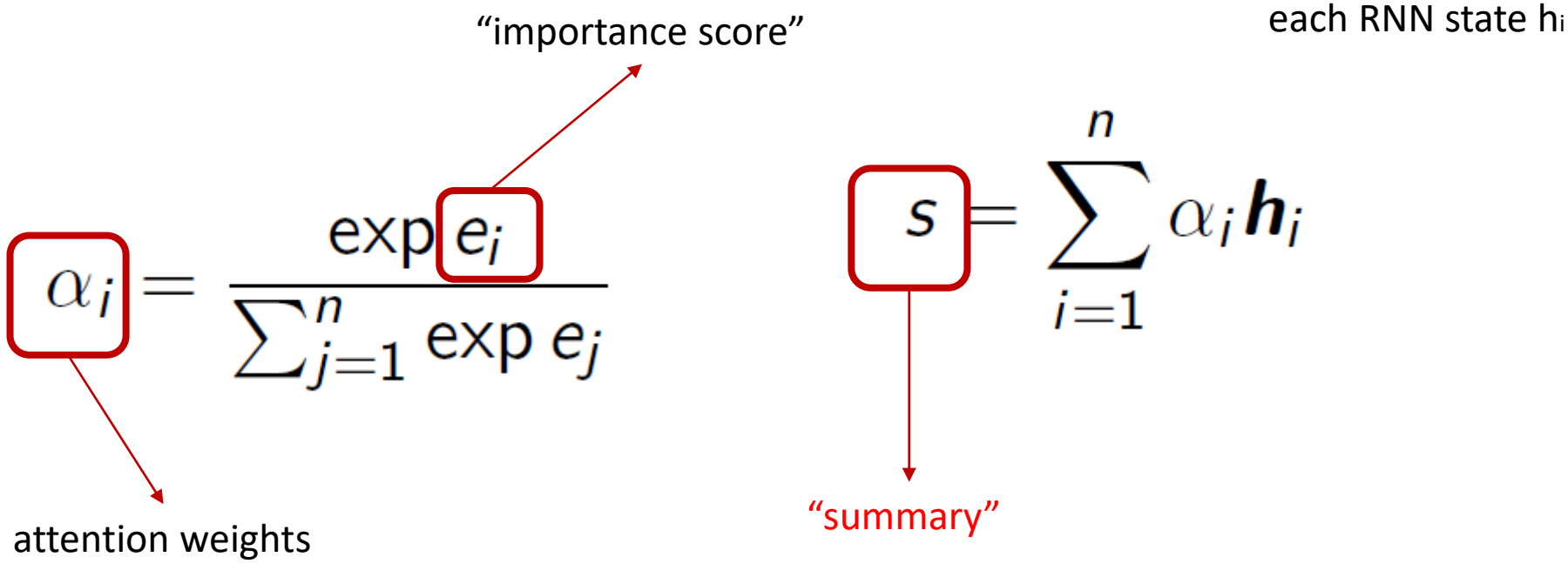


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Neural Machine Translation

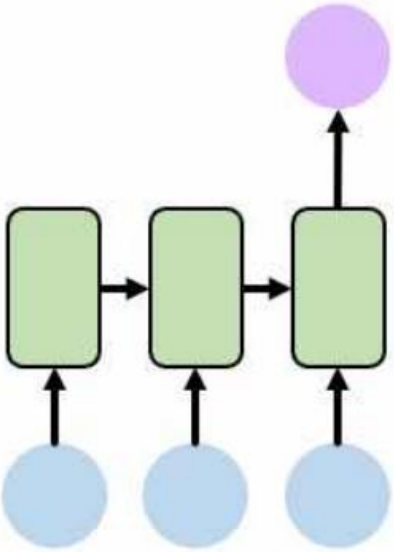
Sequence Modeling Applications

One to One

Binary Classification

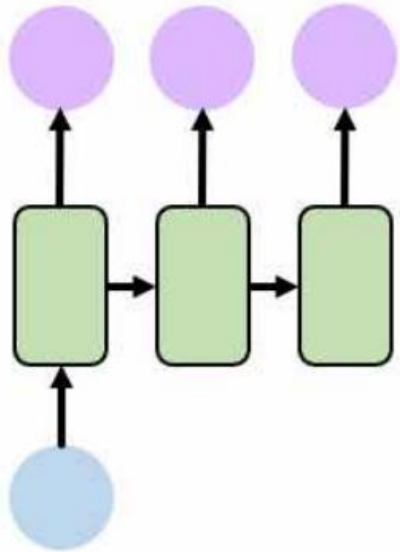
"Will I pass this class?"

Student → Pass?



Many to One

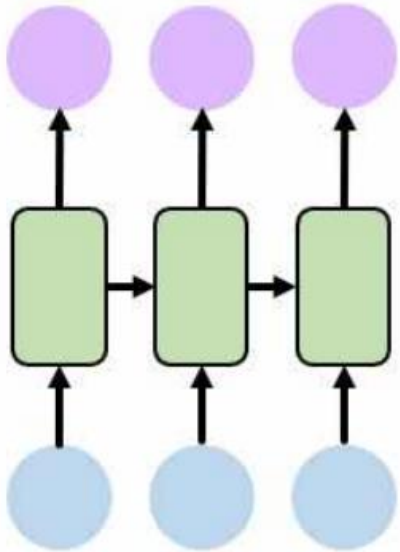
Sentiment Classification



One to Many

Image Captioning

"A baseball player throws a ball."



Many to Many

Machine Translation

What is a Seq2Seq model?

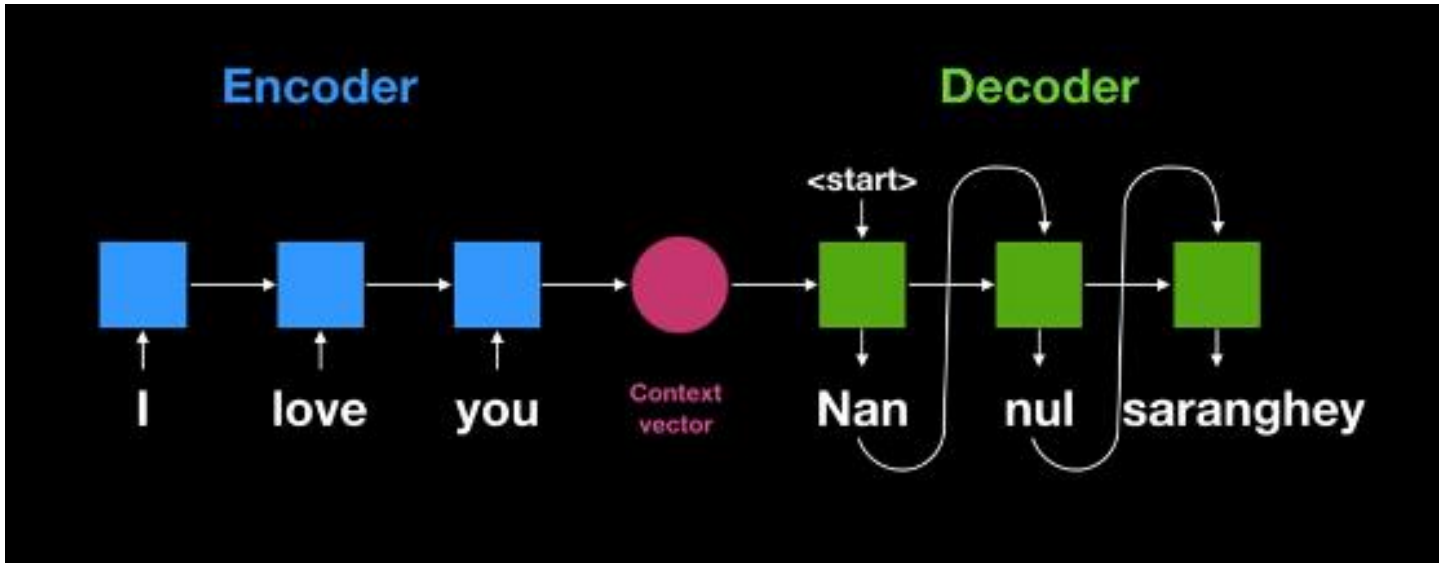
- Sequence-to-sequence learning (Seq2Seq)
  - ✓ Seq2Seq is about training models to convert sequences from one domain (e.g. sentences in English) to sequences in another domain (e.g. the same sentences translated to French).
- This can be used for machine translation

```
"the cat sat on the mat" -> [Seq2Seq model] -> "le chat etait assis sur le tapis"
```

How do I translate the sentence by machine?

- The seq2seq model compresses the input sequence into one fixed-size vector representation, called the context vector, through which the decoder produces the output sequence.

context vector: the final RNN cell states "I love you".



(source) https://www.kaggle.com/code/jeongwonkim10516/attention-mechanism-for-nlp-beginners/notebook

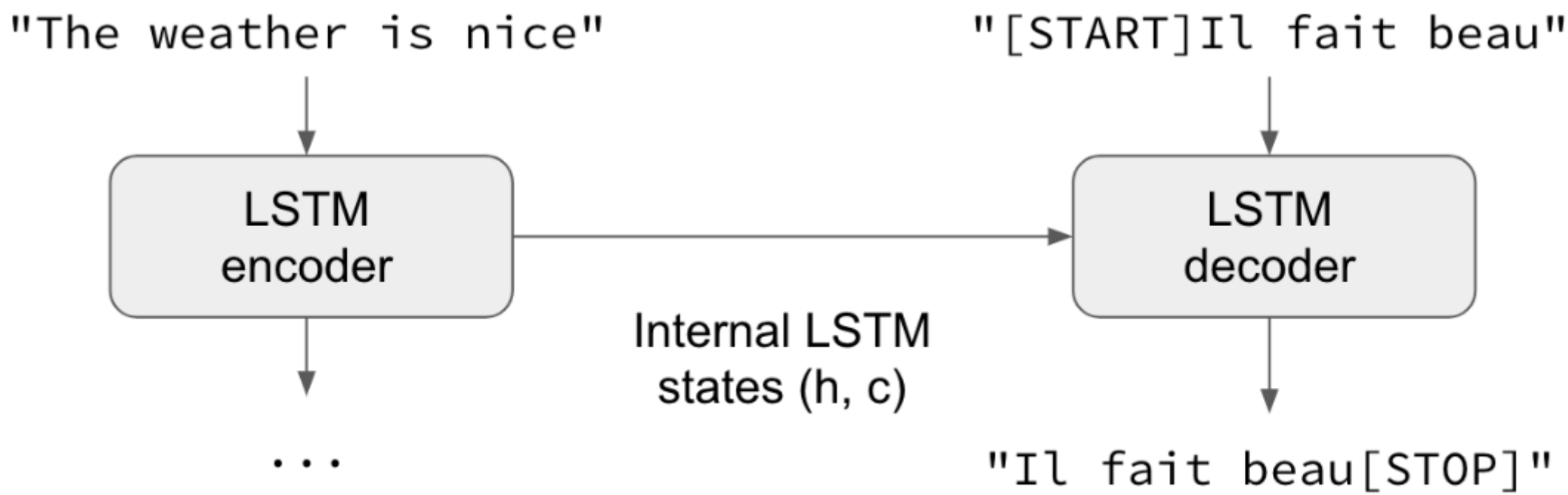
How does a classic Seq2Seq model work?

A Seq2Seq model usually consists of:

- ✓ Encoder: The **encoder** processes all the inputs by transforming them into a single vector, called **context** (usually with a length of 256, 512, or 1024).
- ✓ a **Decoder:** The context contains all the information that the encoder was able to detect from the input.
- ✓ a **Context** (*vector*): the vector is sent to the **decoder** which formulates the output sequence.

The general case of machine translation

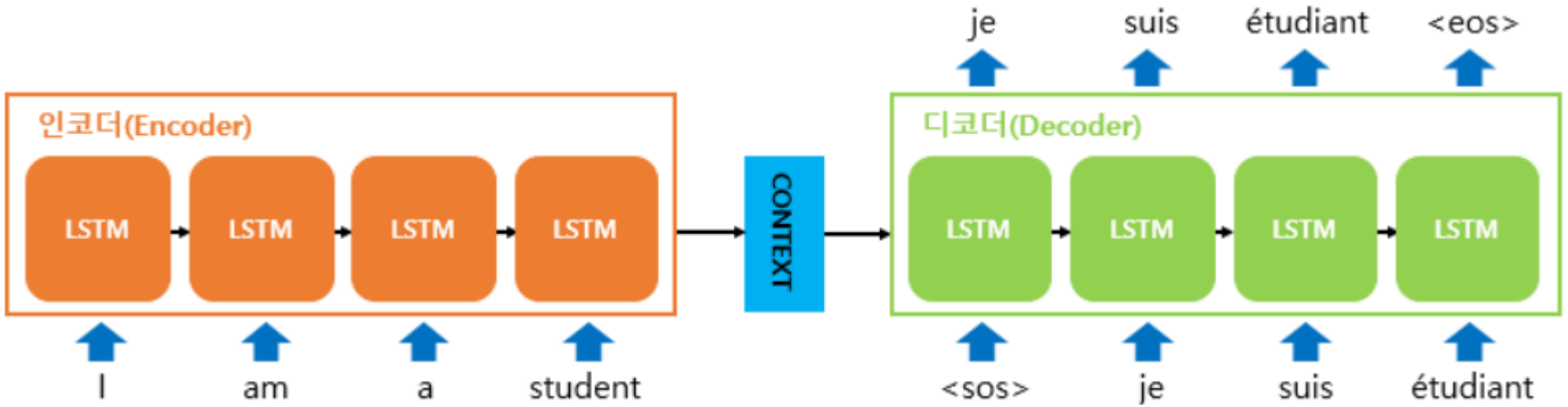
- Input sequences and output sequences have different lengths



Encoder and decoder are RNN architectures.

❖ context vector

✓ The context vector is the first hidden state of the decoder RNN cell

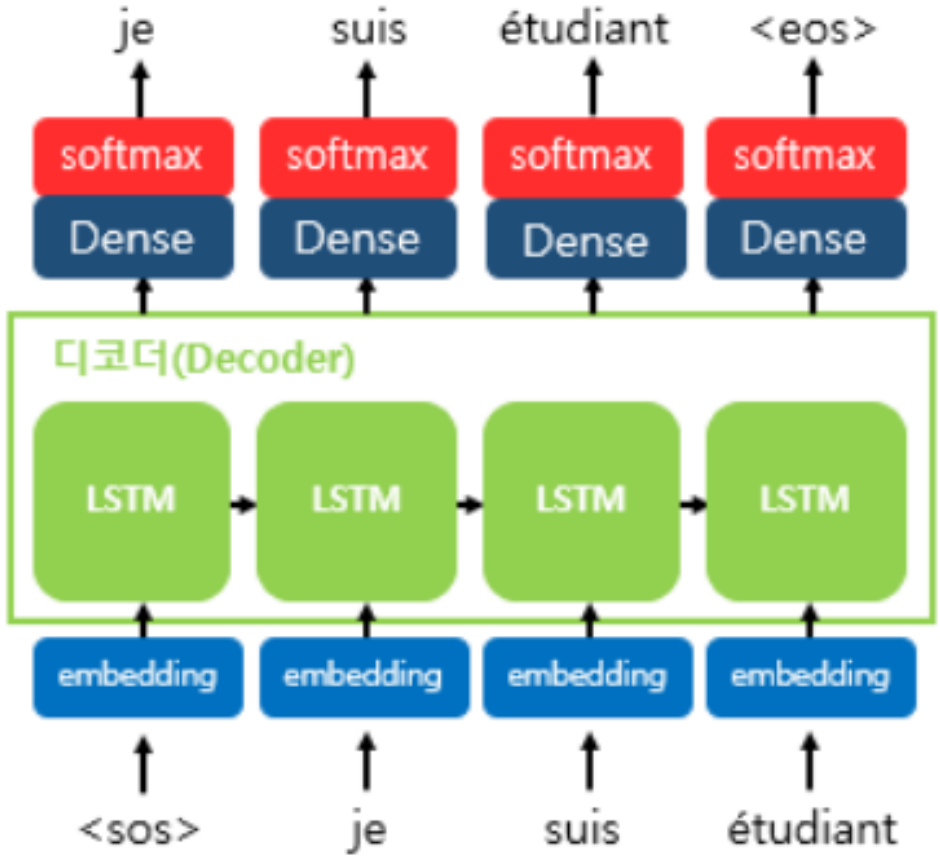


✓ Decoder is essentially an RNNLM (RNN Language Model)

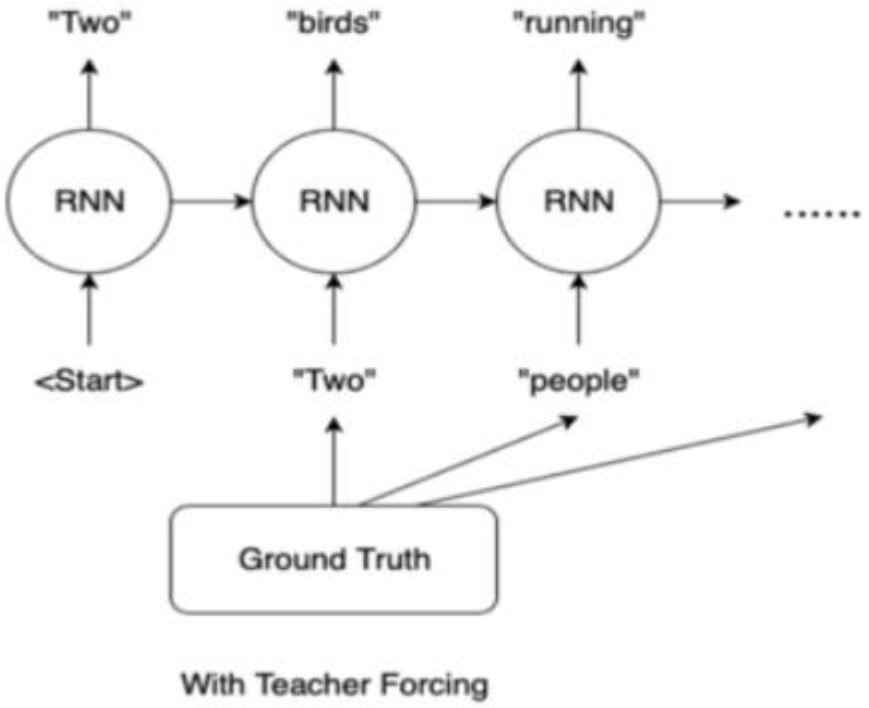
- Many-to-Many

Seq2Seq: Decoder part

- Softmax for next prediction word

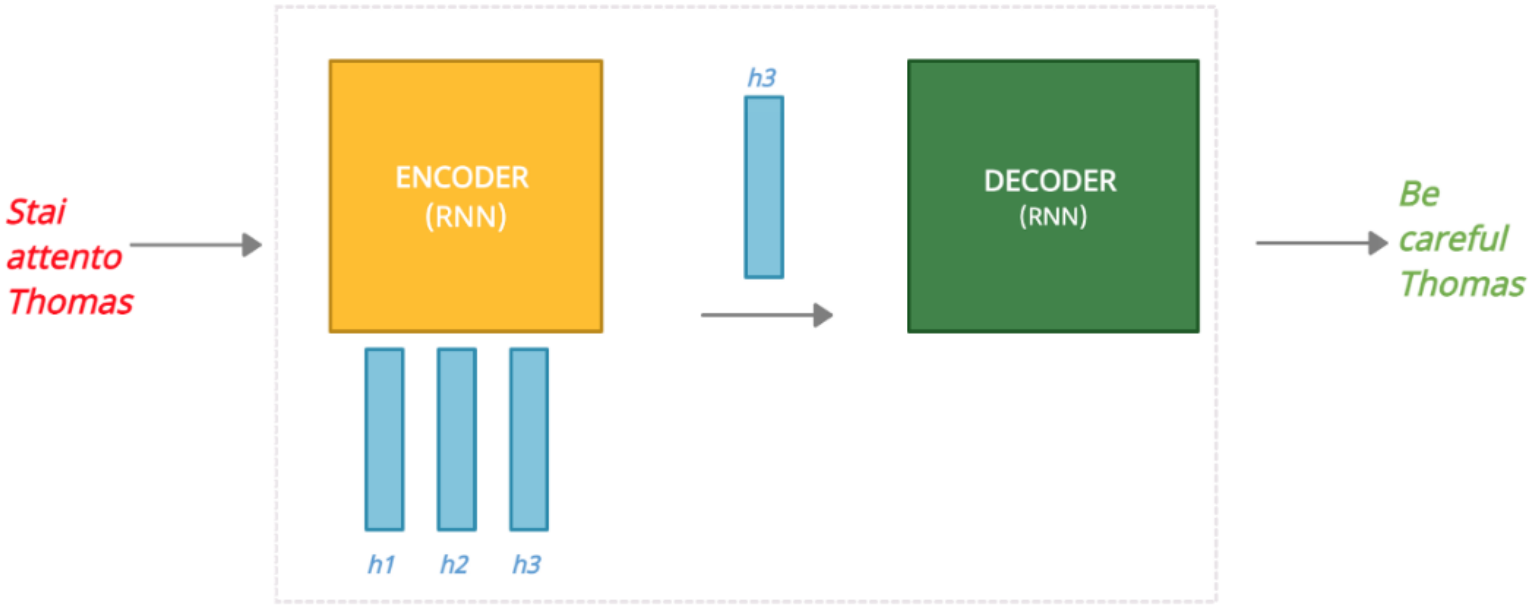


Teacher Forecing Learning



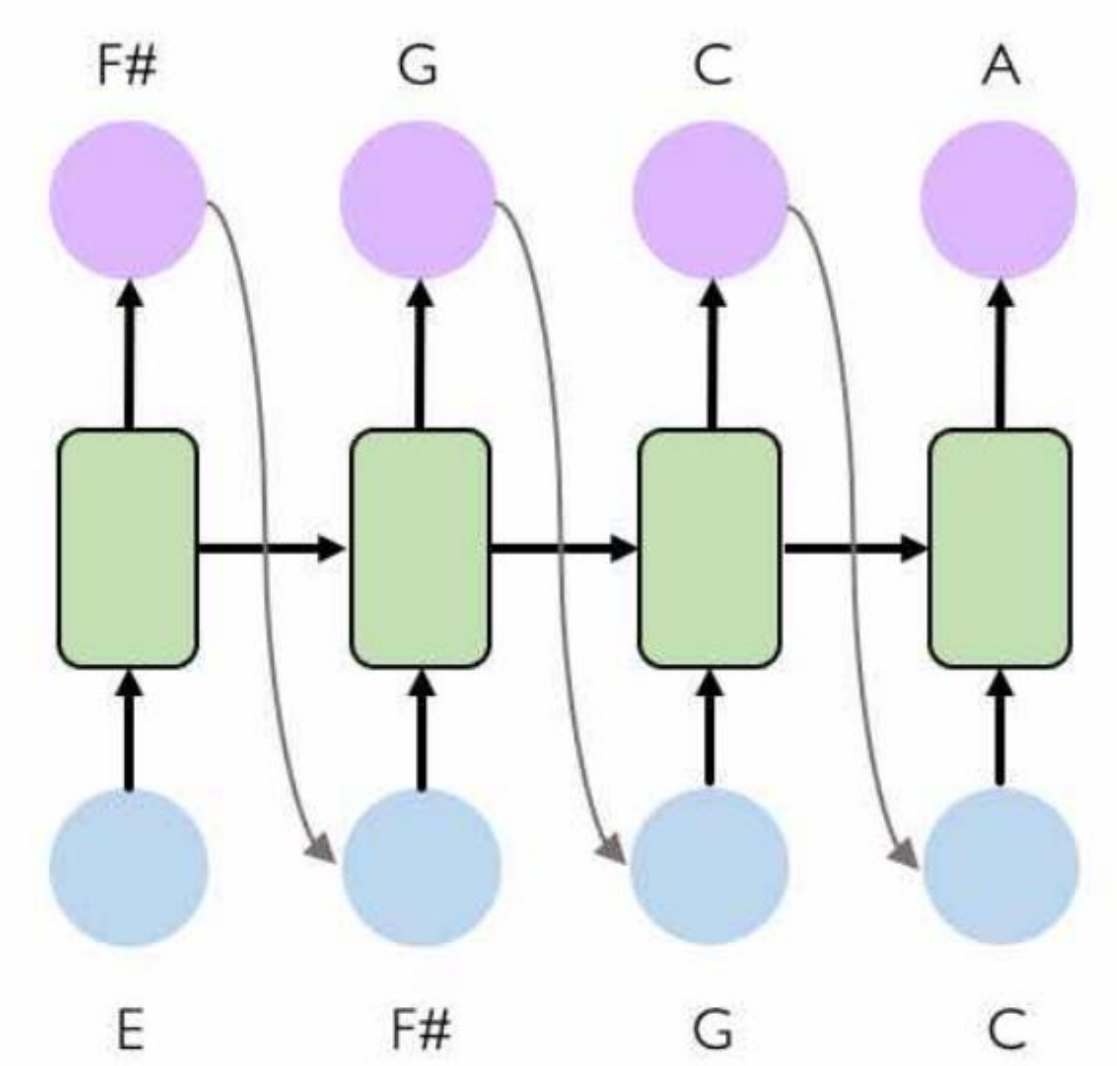
Limitation of Seq2Seq model

- Main problem with seq2seq models
  - ✓compress all the information into one fixed-size vector results in information loss.
- This is the problem that attention solves!
  - ✓The last **hidden state (*h3*)** becomes the content that is sent to the decoder
  - ✓the encoder is “forced” to send only a single vector, regardless of the length of our input

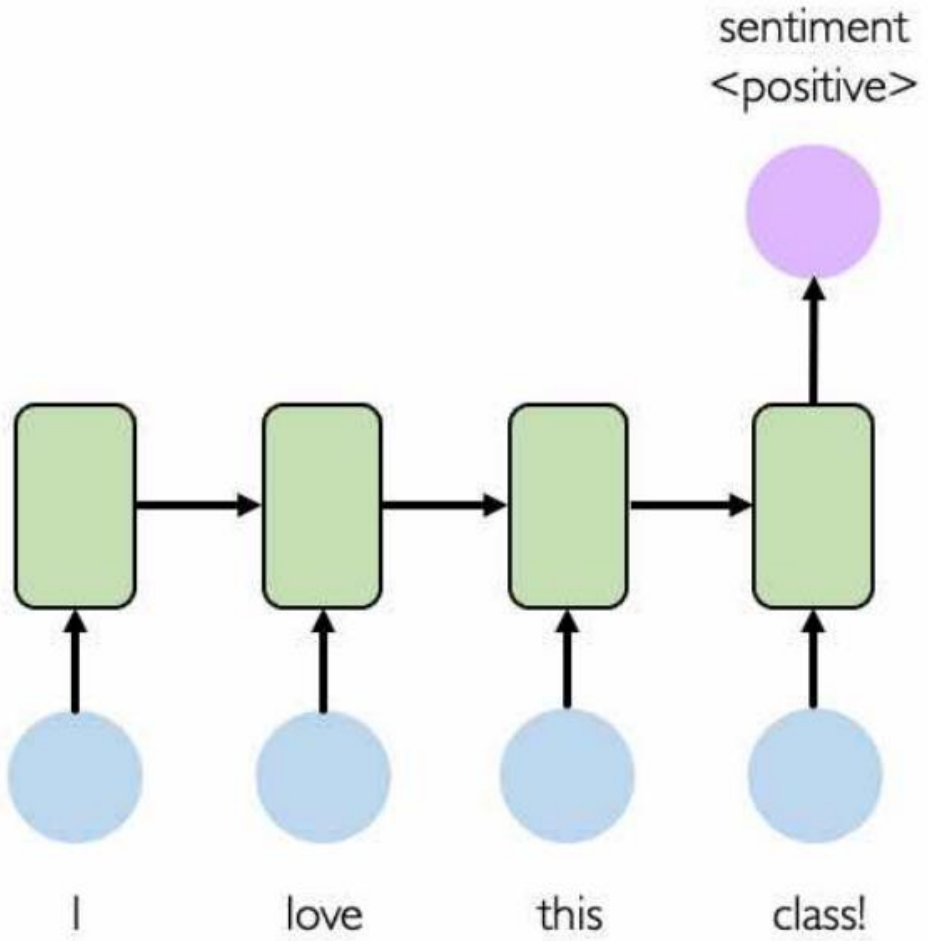


RNN Applications and Limitations

Example Task : Music Generation



Example Task : Sentiment Classification

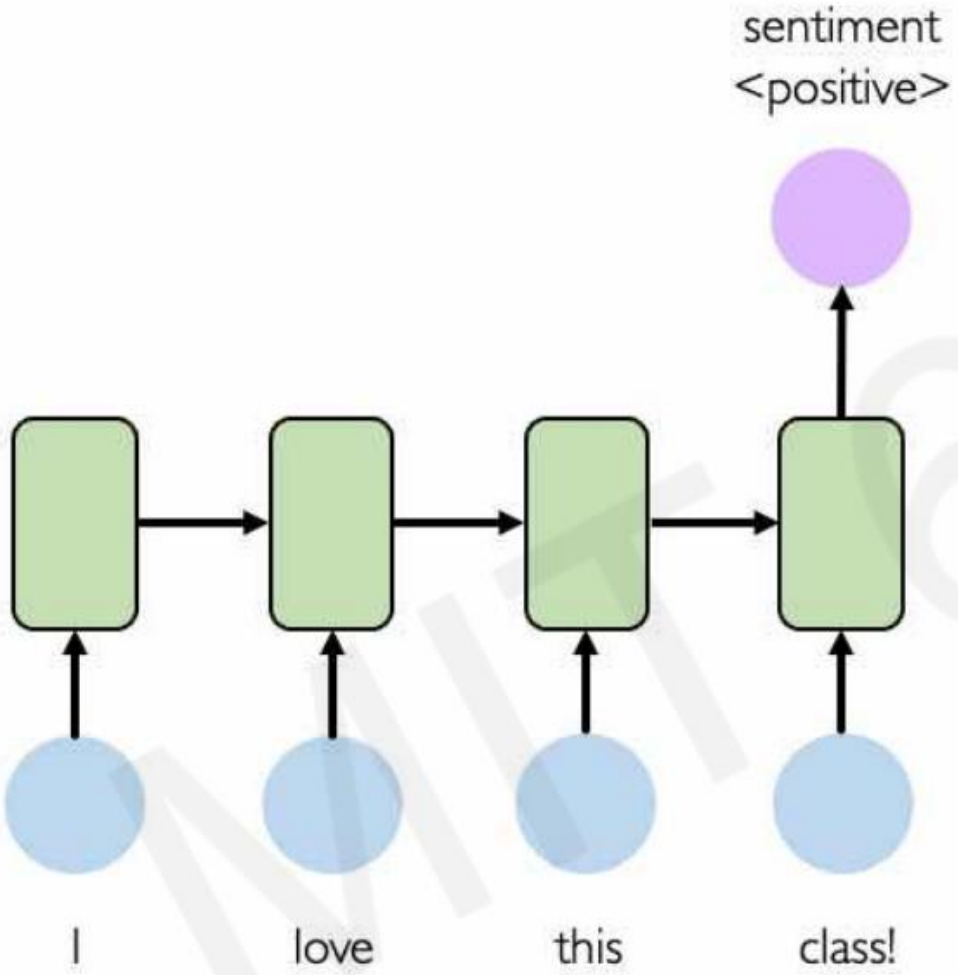


Input: sequence of words

Output: probability of having positive sentiment

```
loss = tf.nn.softmax_cross_entropy_with_logits(y, predicted)
```

Example Task : Sentiment Classification



Tweet sentiment classification

Ivar Hagendoorn
@IvarHagendoorn
Follow

The @MIT Introduction to #DeepLearning is definitely one of the best courses of its kind currently available online introtodeeplearning.com

12:45 PM - 12 Feb 2018

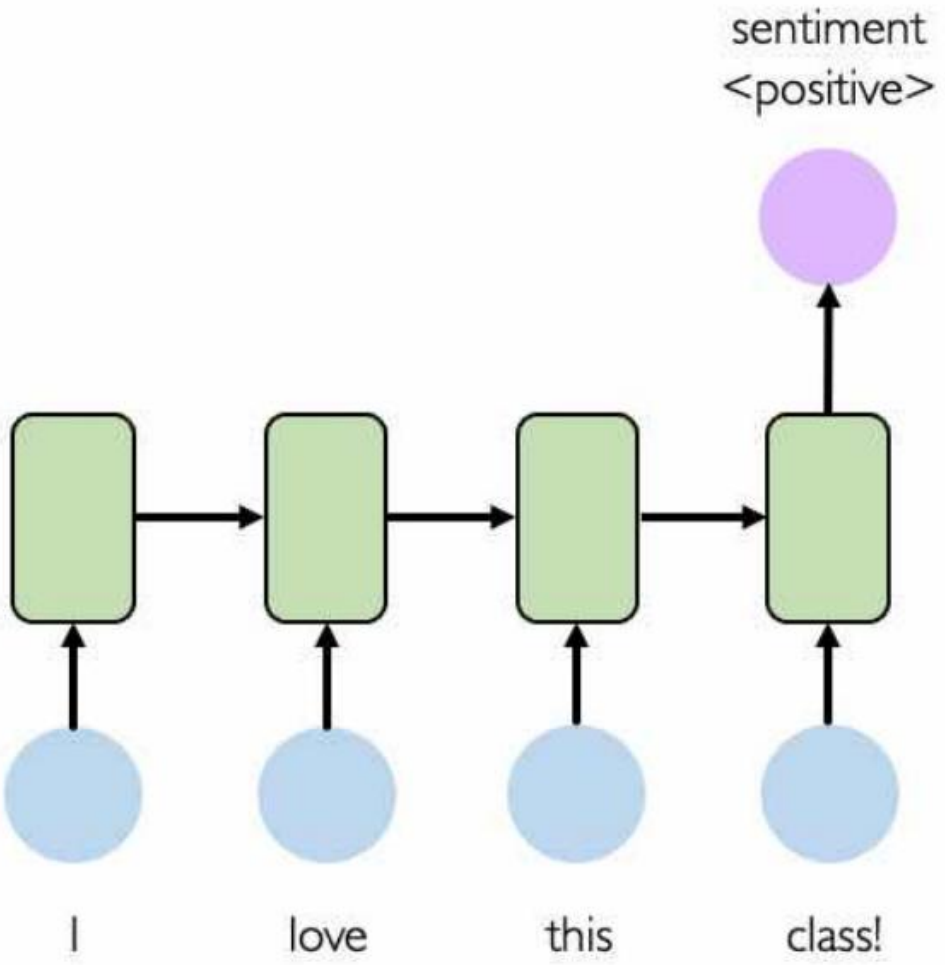
Angels-Cave
@AngelsCave
Follow

Replying to @Kazuki2048

I wouldn't mind a bit of snow right now. We haven't had any in my bit of the Midlands this winter! :(

2:19 AM - 25 Jan 2019

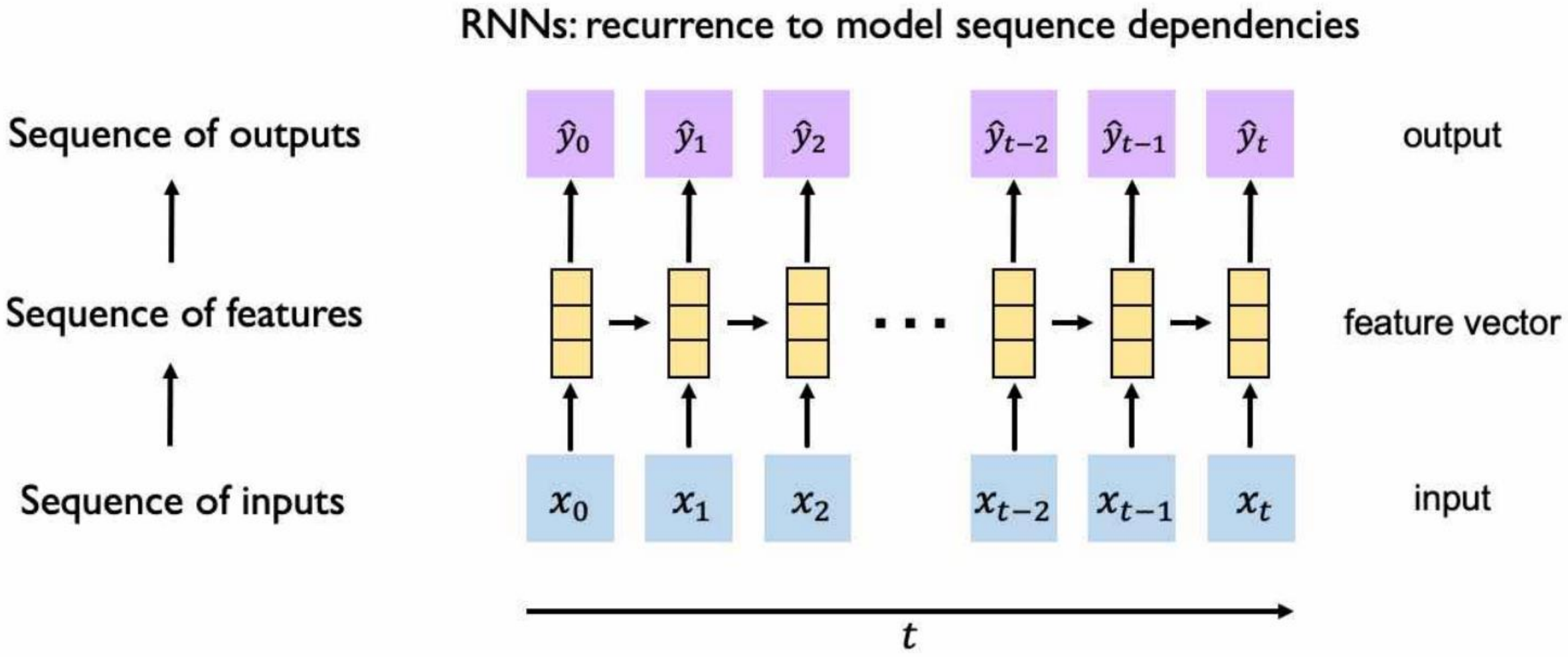
Limitations fo Recurrnet Models



Limitations of RNNs

- Encoding bottleneck
- Slow, no parallelization
- Not long memory

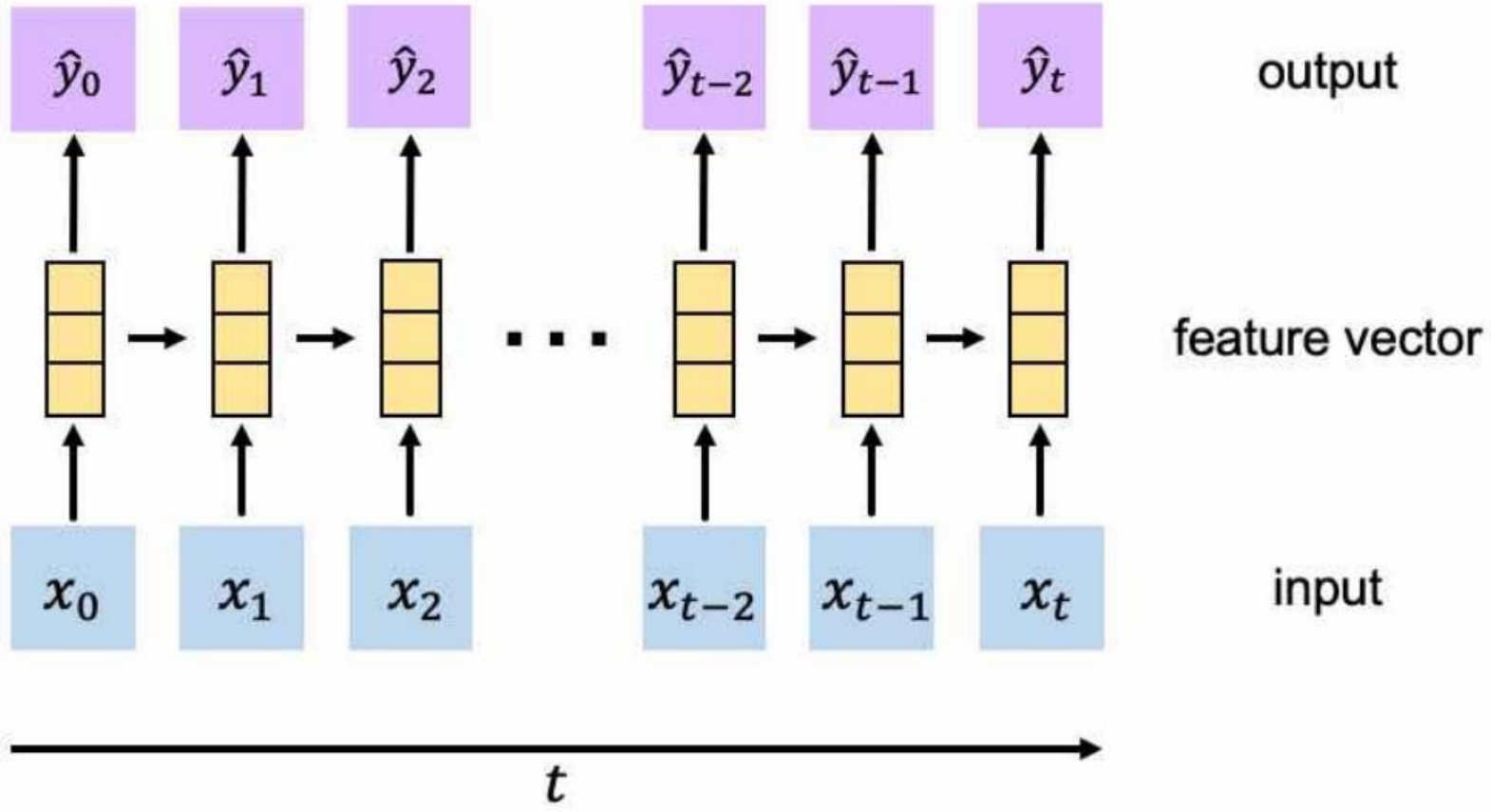
Goal of Sequence Modeling



Goal of Sequence Modeling

Limitations of RNNs

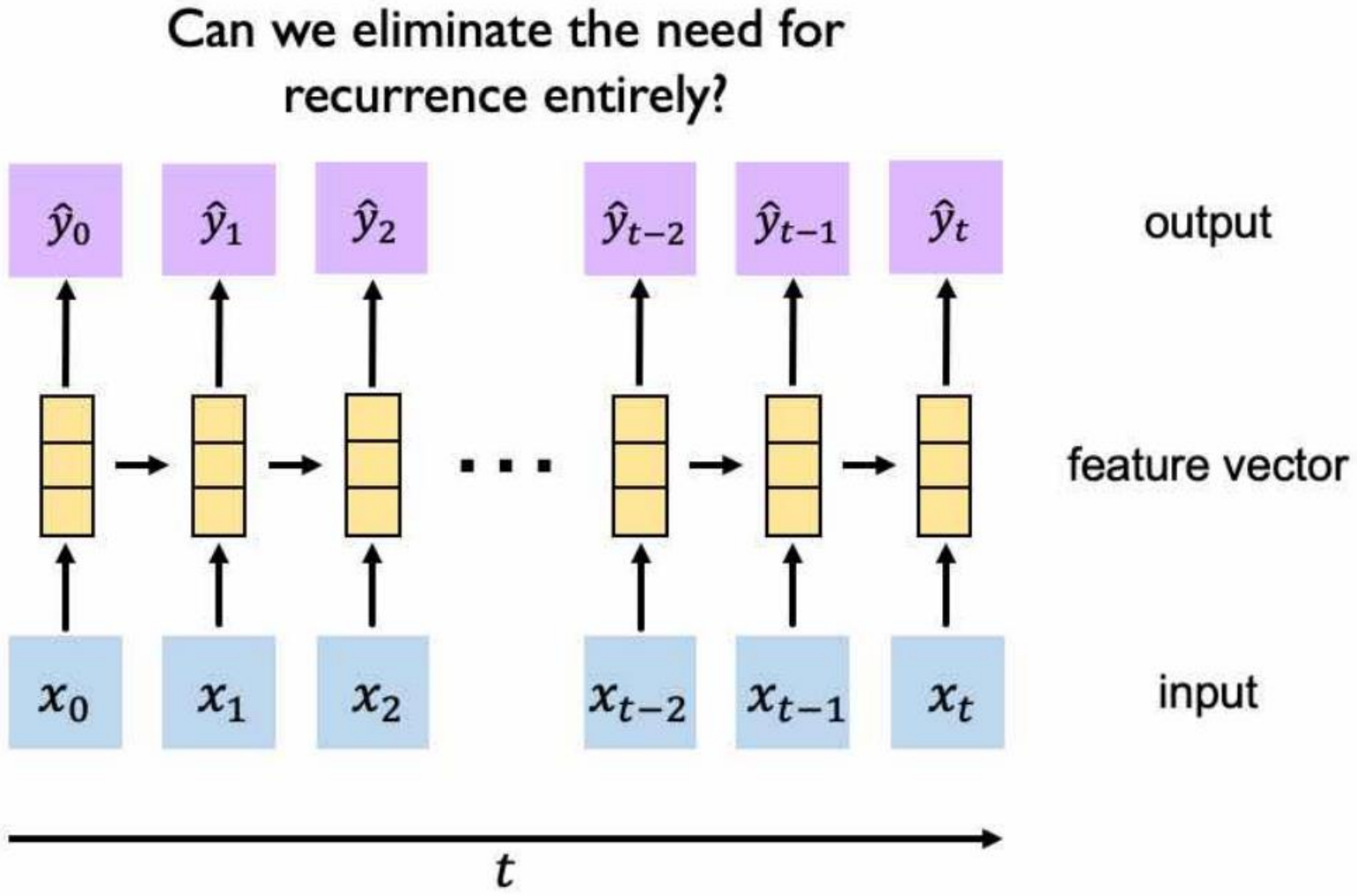
- Encoding bottleneck
- Slow, no parallelization
- Not long memory



Goal of Sequence Modeling

Desired Capabilities

- Continuous stream
- Parallelization
- Long memory

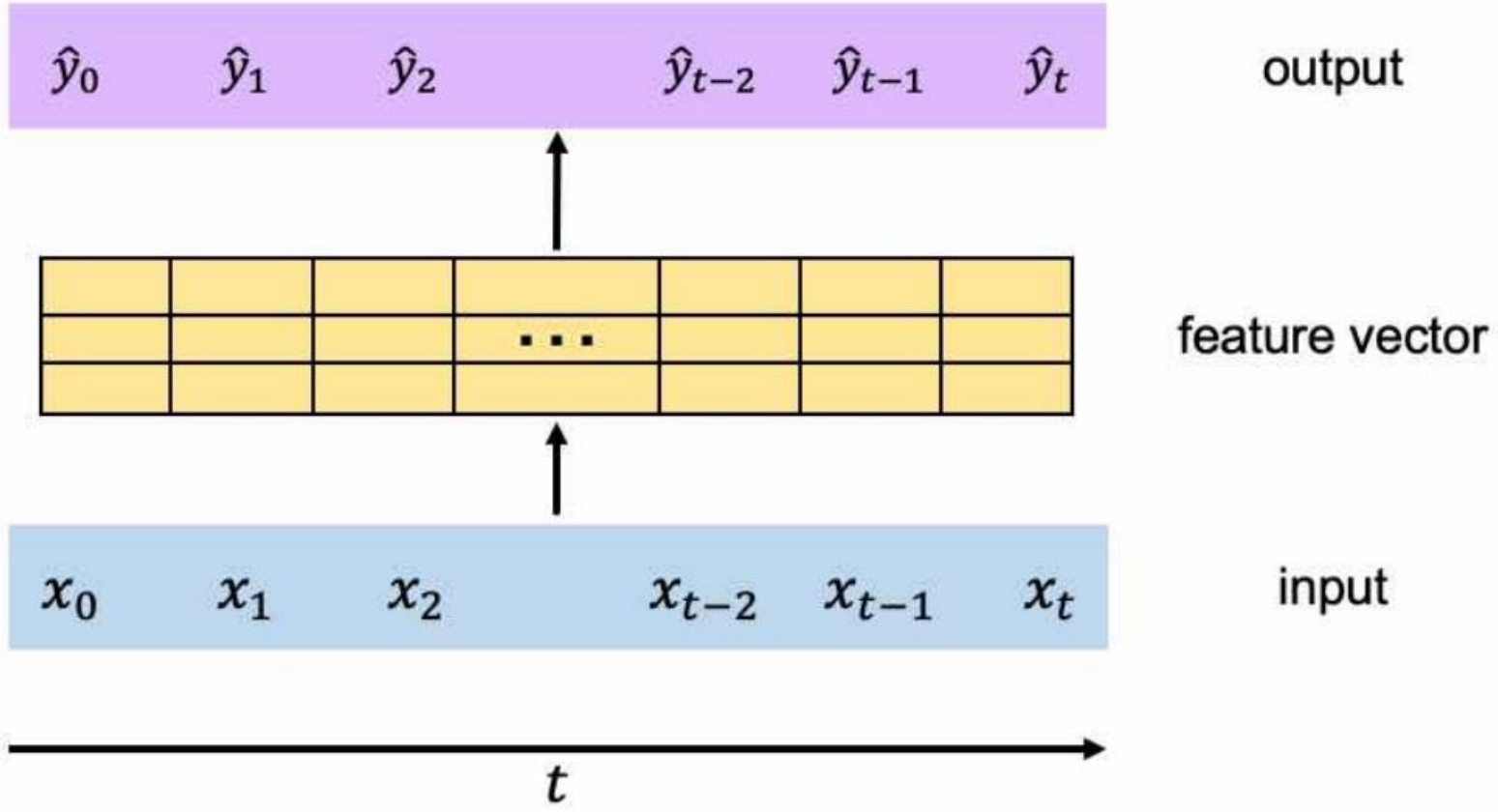


Goal of Sequence Modeling

Desired Capabilities

- Continuous stream
- Parallelization
- Long memory

Can we eliminate the need for recurrence entirely?



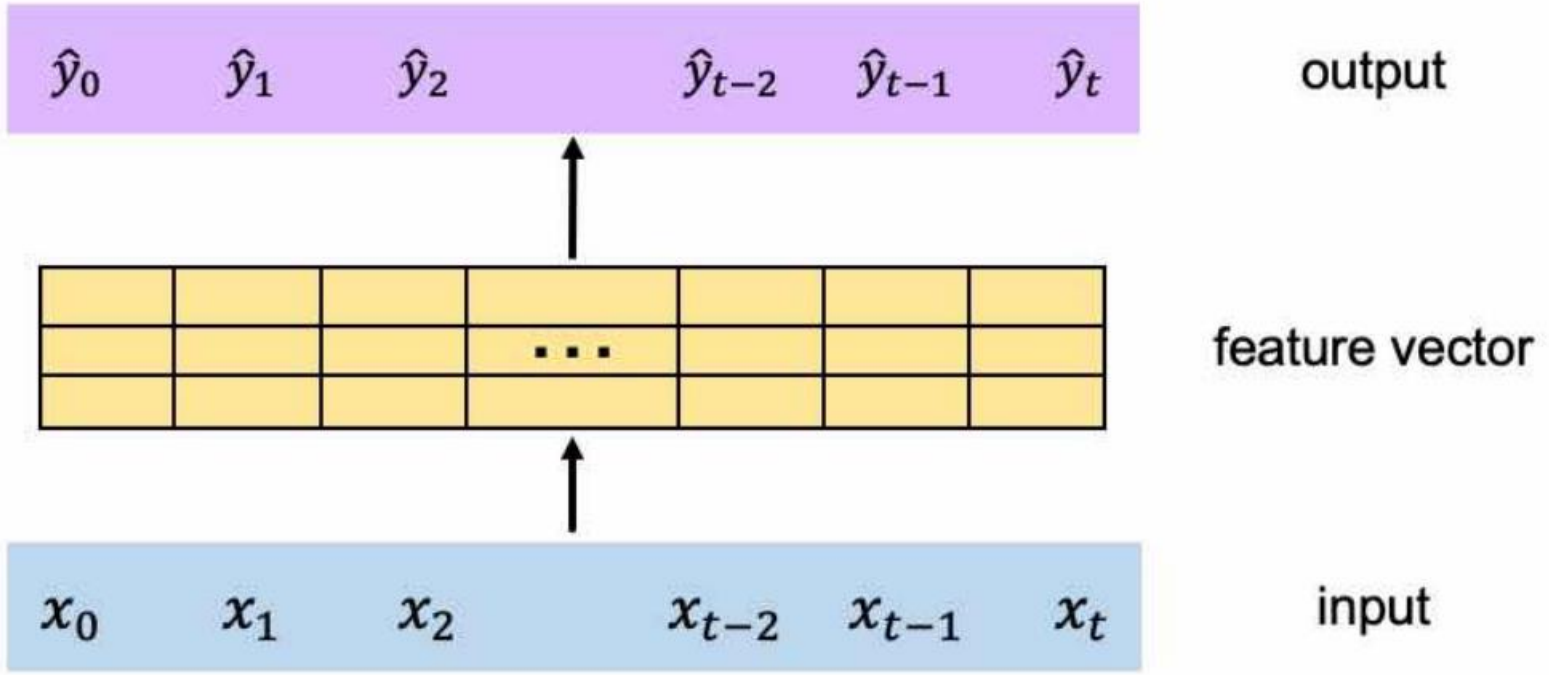
Goal of Sequence Modeling

Idea 1: Feed everything into dense network

- ✓ No recurrence
- ✗ Not scalable
- ✗ No order
- ✗ No long memory

Idea: Identify and attend to what's important

Can we eliminate the need for recurrence entirely?



Attnetion Mechanism

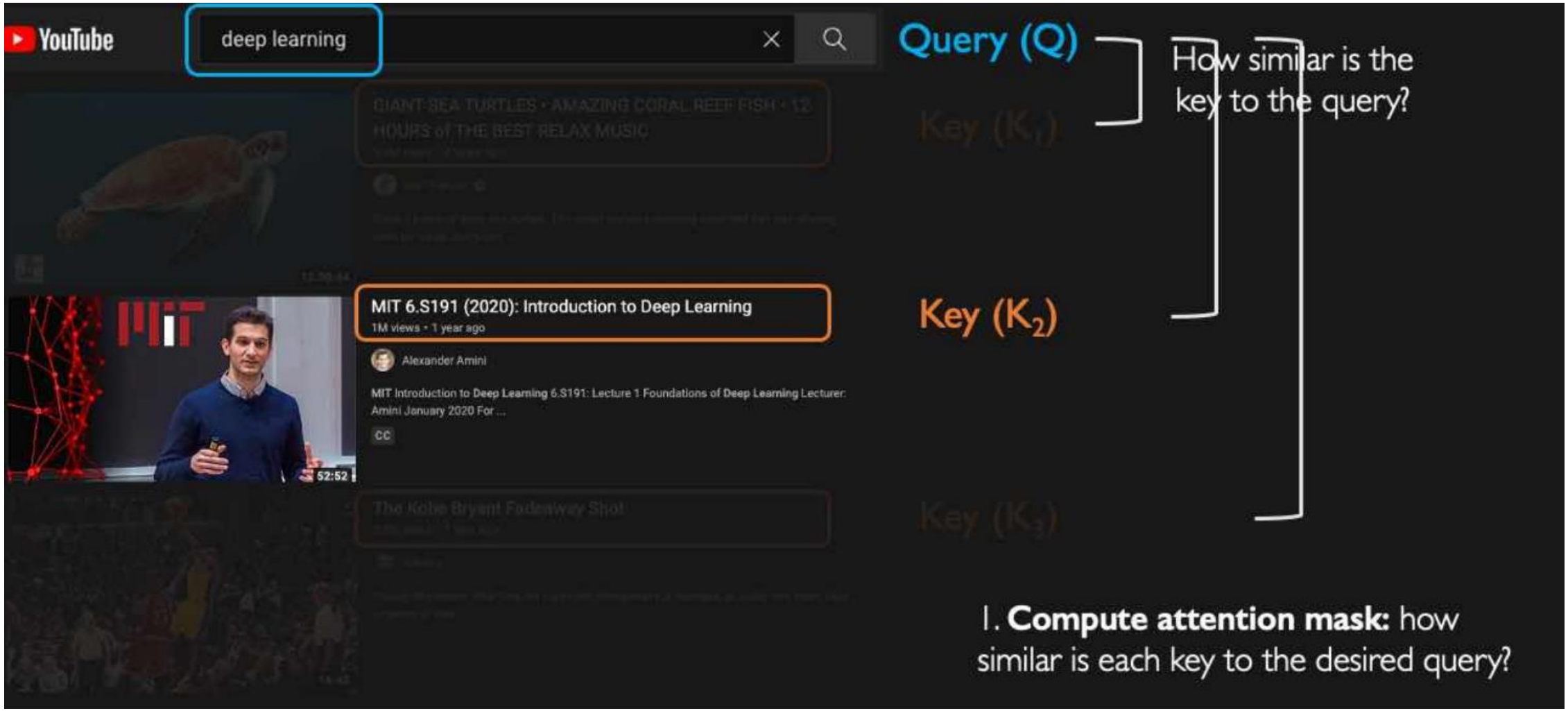
Intution Behind Self-Attention

Attending to the most important parts of an input.

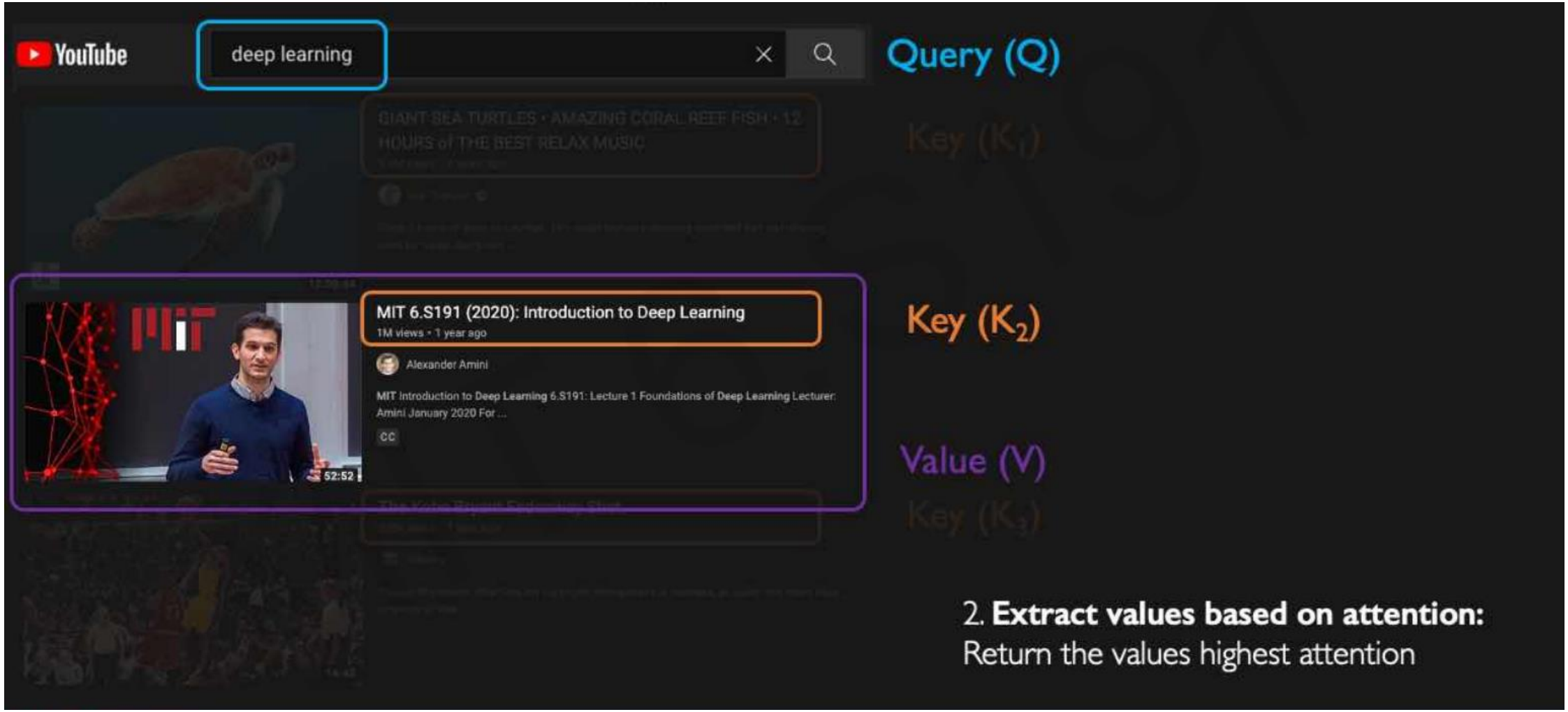
1. Identify which parts to attend to
2. Extract the features with high attention

Similar to a search problem!

Understanding Attention with Search

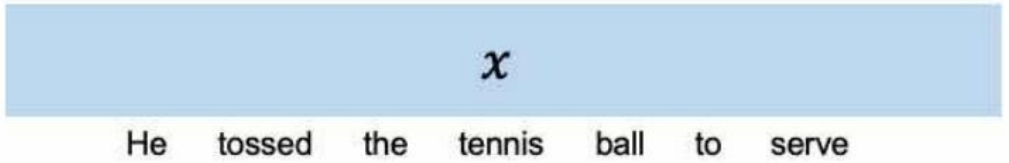


Understanding Attention with Search



Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.



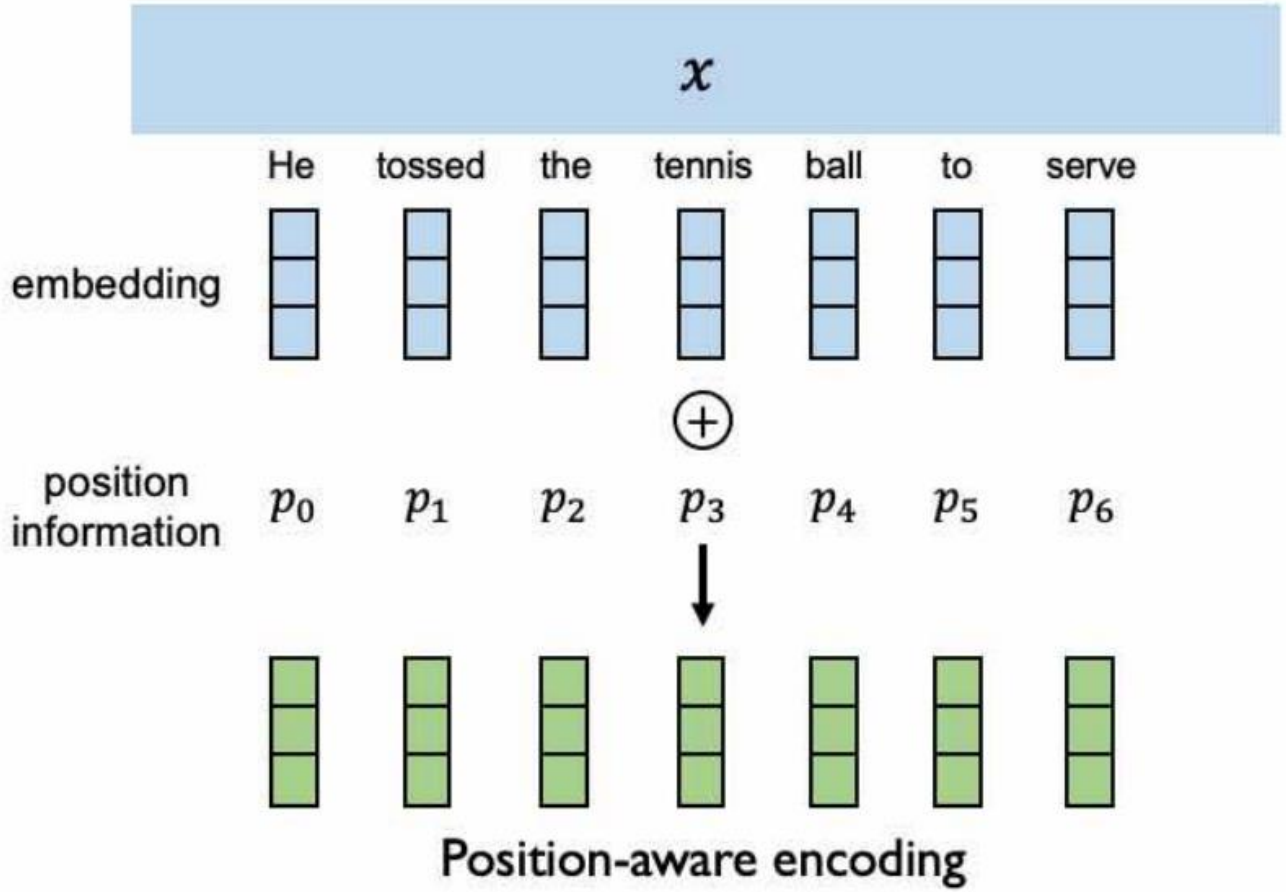
1. Encode **position** information
2. Extract query, key, value for search
3. Compute attention weighting
4. Extract features with high attention

Data is fed in all at once! Need to encode position information to understand order.

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract query, key, value for search
3. Compute attention weighting
4. Extract features with high attention

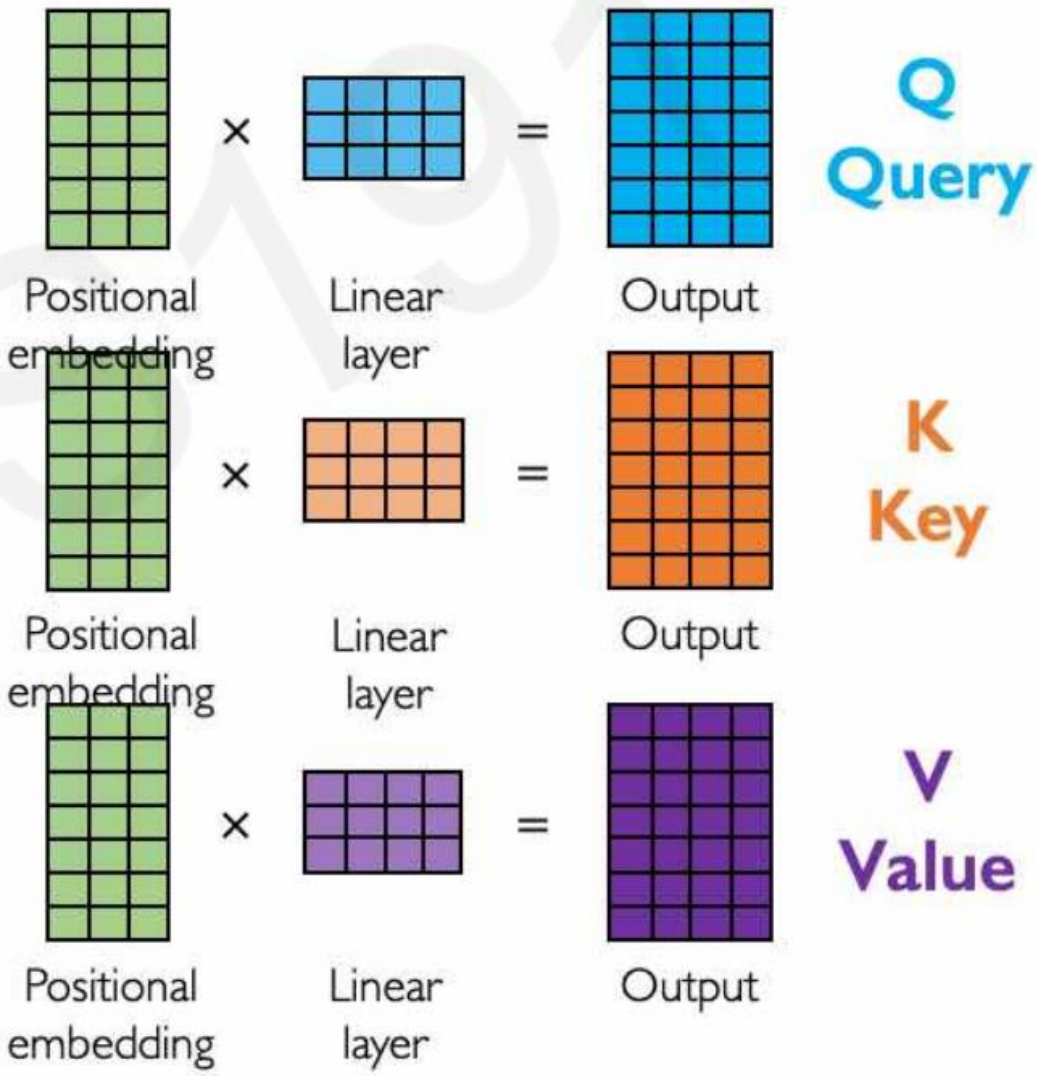


Data is fed in all at once! Need to encode position information to understand order.

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute attention weighting
4. Extract features with high attention



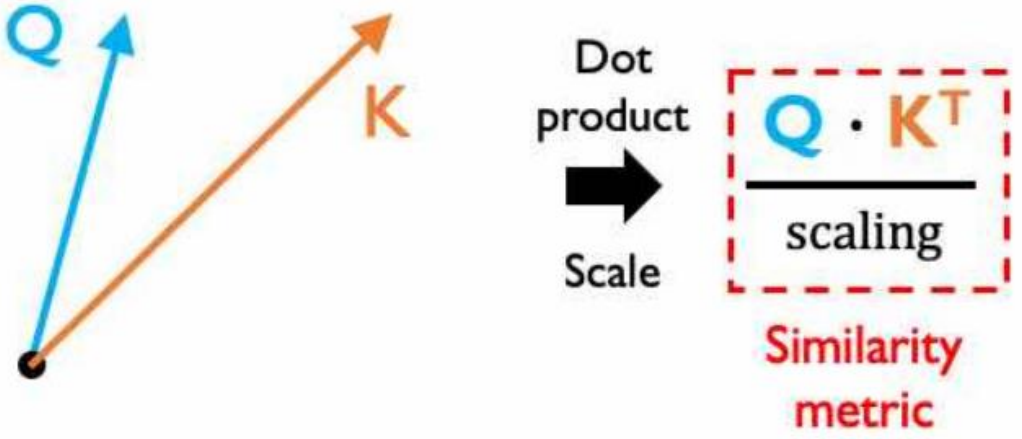
Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract features with high attention

Attention score: compute pairwise similarity between each query and key

How to compute similarity between two sets of features?



Also known as the "cosine similarity"

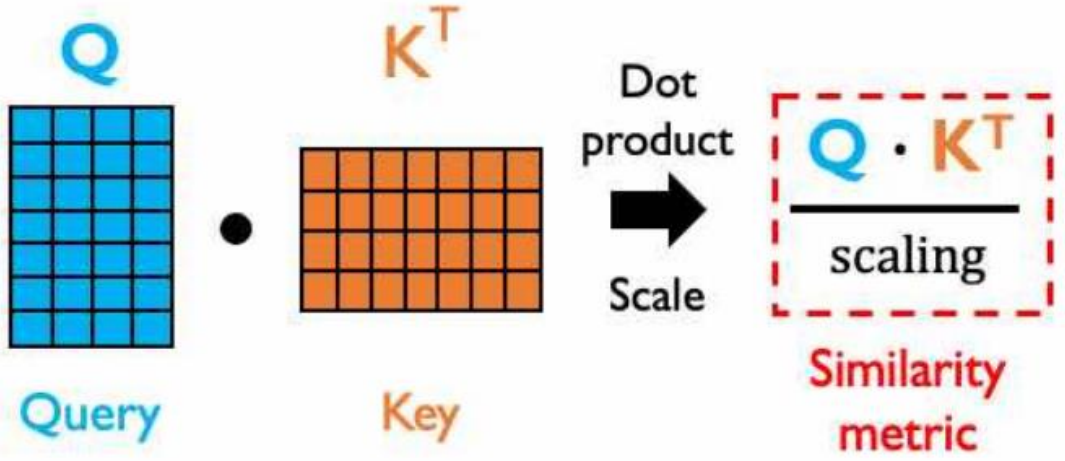
Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract features with high attention

Attention score: compute pairwise similarity between each query and key

How to compute similarity between two sets of features?



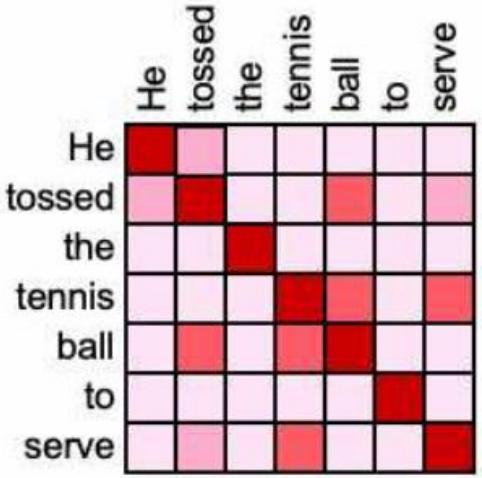
Also known as the "cosine similarity"

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract features with high attention

Attention weighting: where to attend to! How similar is the key to the query?



$$softmax\left(\frac{Q \cdot K^T}{scaling}\right)$$

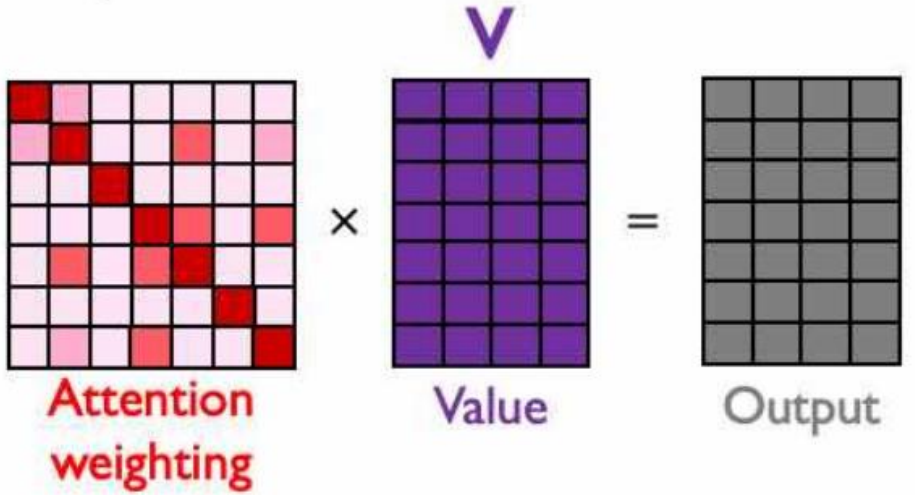
Attention weighting

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract **features with high attention**

Last step: self-attend to extract features



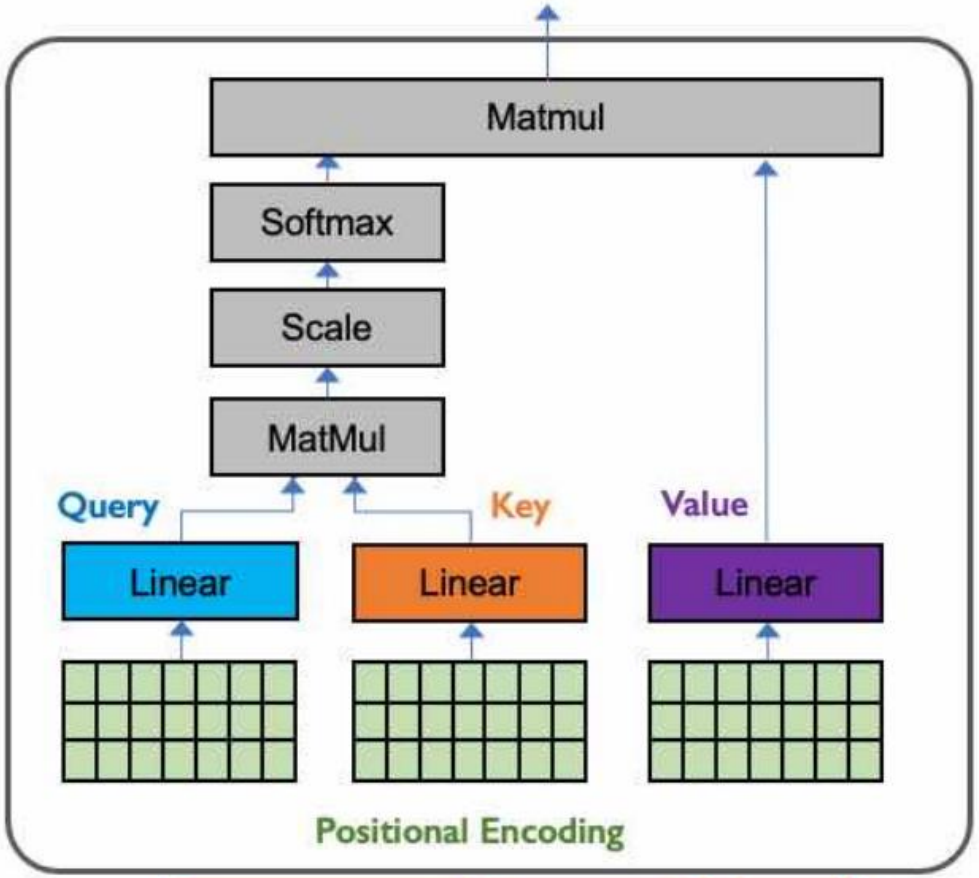
$$softmax\left(\frac{Q \cdot K^T}{scaling}\right) \cdot V = A(Q, K, V)$$

Learning Self-Attention with Neural Networks

Goal: identify and attend to most important features in input.

1. Encode **position** information
2. Extract **query, key, value** for search
3. Compute **attention weighting**
4. Extract **features with high attention**

These operations form a self-attention head that can plug into a larger network. Each head attends to a different part of input.



$$softmax\left(\frac{Q \cdot K^T}{scaling}\right) \cdot V$$

Self-Attention Applied

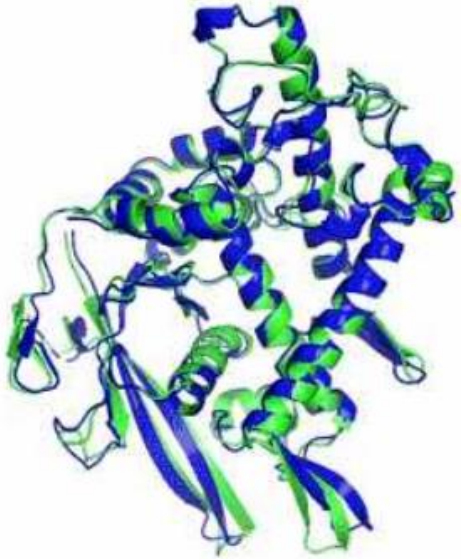
Language Processing

An armchair in the shape of an avocado

BERT, GPT-3

Devlin et al., *NAACL* 2019
Brown et al., *NeurIPS* 2020

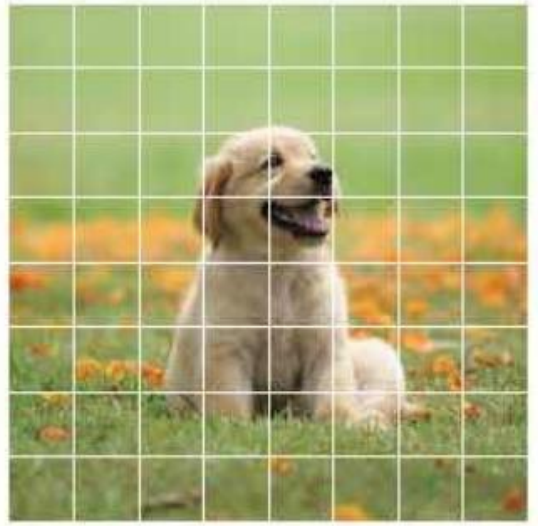
Biological Sequences



AlphaFold2

Jumper et al., *Nature* 2021

Computer Vision



Vision Transformers

Dosovitskiy et al., *ICLR* 2020

Transformer Model for Traffic Flow Prediction

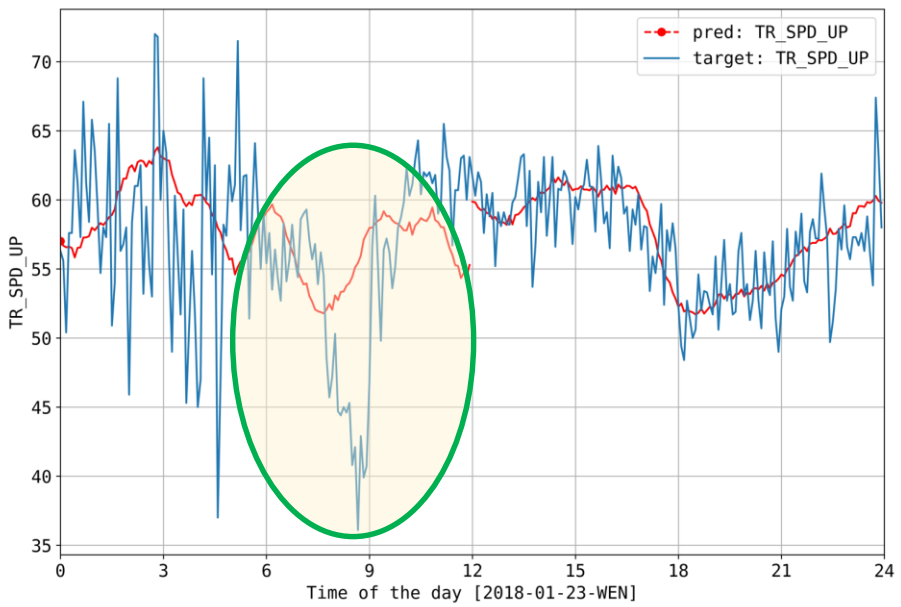
대상 도로 : 대학로 (유성구청네거리-구성삼거리)

❖ 약 1.4km 왕복 4차로, RSE 3개, 영상검지기 5개(현재 3개, 향후 2개)

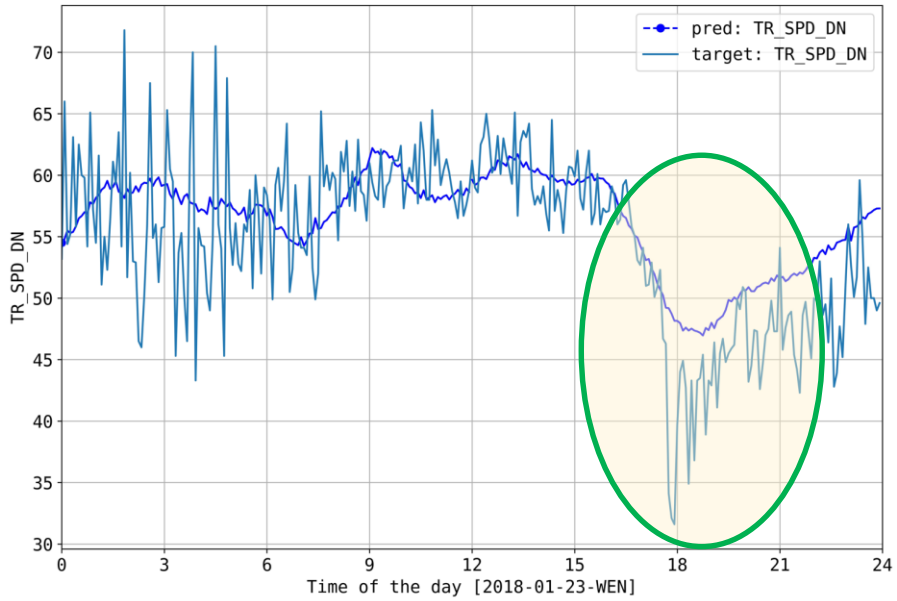
LSTM을 이용한 교통 흐름 예측 및 새로운 모델의 필요

LSTM 예측은 RNN에 비하여 성능은 좋은나, Long-Sequence 장기예측에는 성능저하가 발생한다.

상행 12시간예측 (2018.01.23.,수요일)



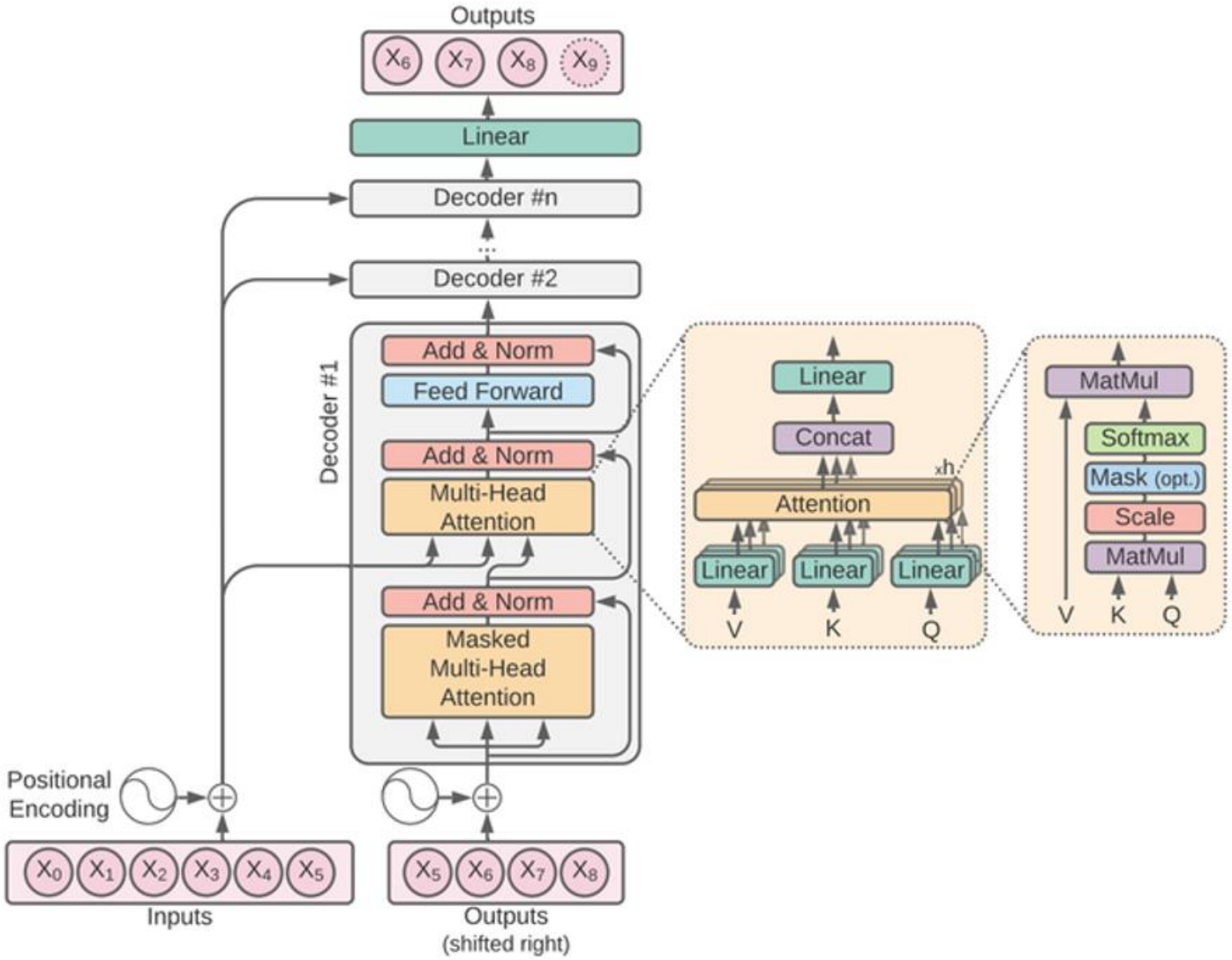
하행 12시간예측 (2018.01.23.,수요일)



LSTM 장기예측(12시간)과 예측 정확은 재고할 필요가 있다. → 새로운 모델이 필요하다.

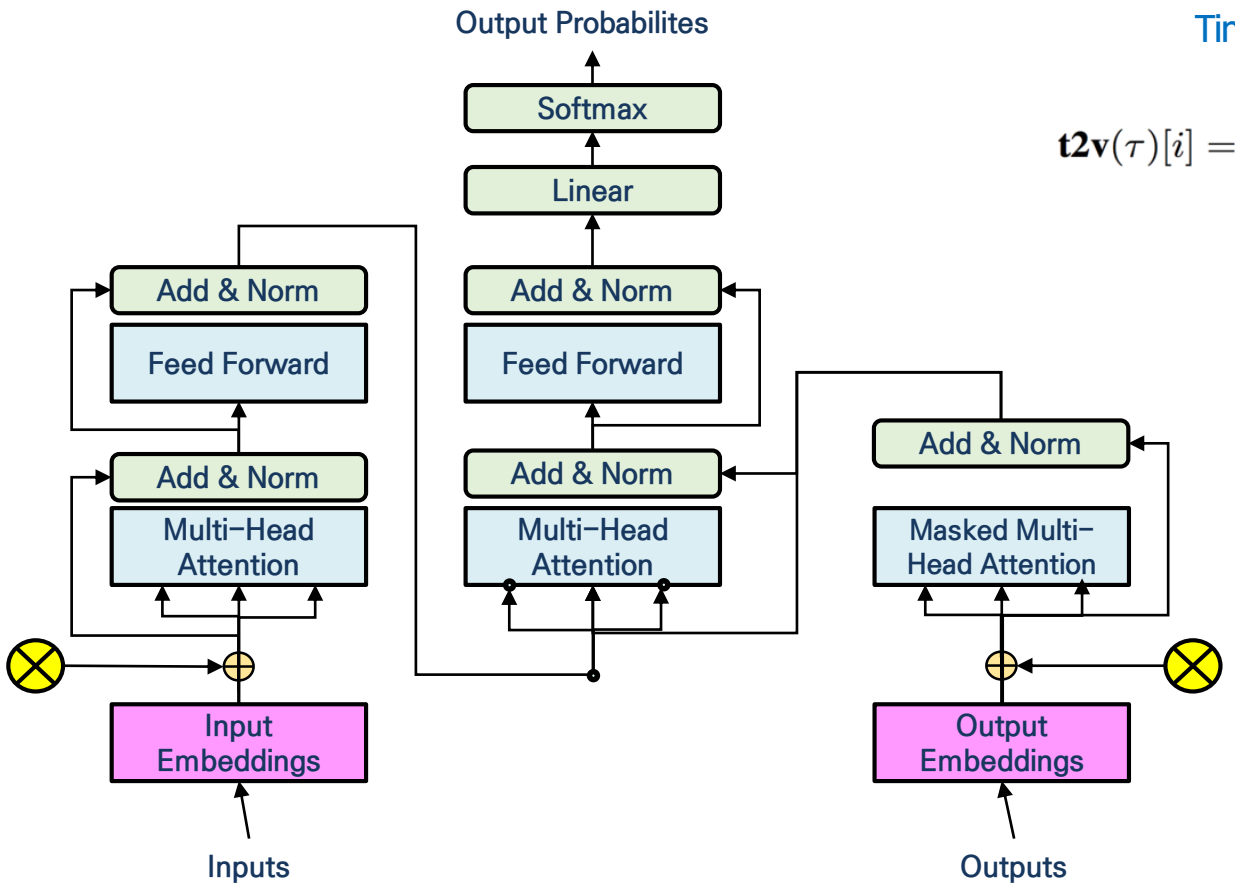
☆ LSTM 장기 예측(12시간) 결과는 상대적으로 정확도가 현저히 떨어진다.

Multi-head Attention Mechaism



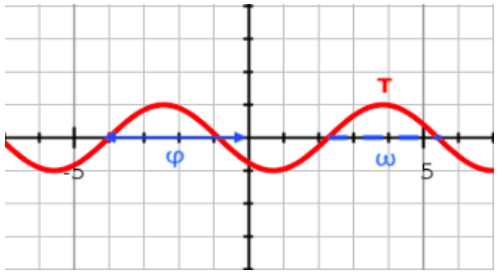
Position Encoding : Time2Vec

자연어 처리 트랜스포머 모델의 포지셔널 인코딩을 교통에 적합하게 Time2Vec 개발이 핵심이다.



❖ Attention 도입으로 시계열성을 없어지고, 대신에 Time2Vec 임베딩을 이용하여 시공간적인 정보 유지

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i \tau + \varphi_i, & \text{if } i = 0. \\ \mathcal{F}(\omega_i \tau + \varphi_i), & \text{if } 1 \leq i \leq k. \end{cases}$$



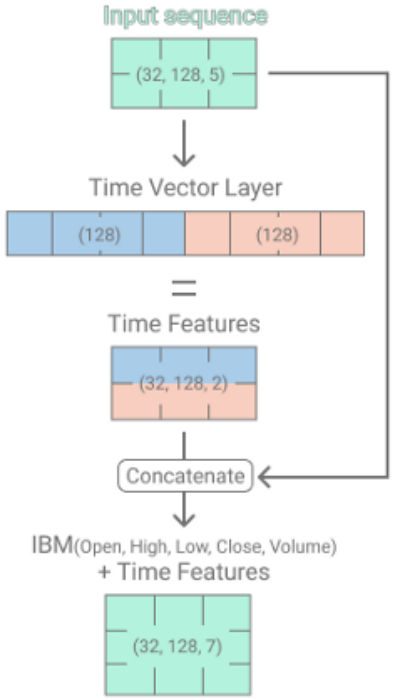
주기성: 삼각함수(sin)를 사용

비주기성: 선형함수

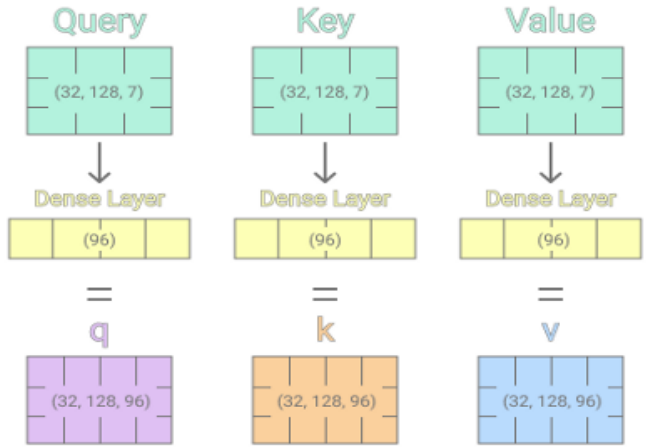
Attention : Query, Key, Value

Time2Vec

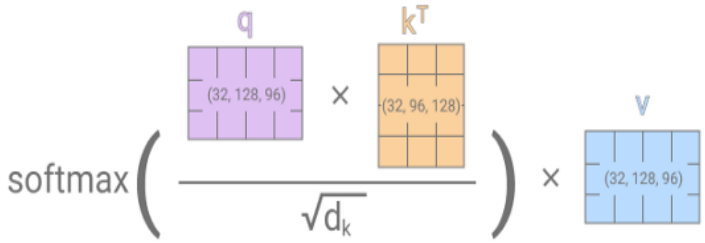
- 5개 VDS 17, VDS 18, VDS 19, VDS 20, VDS 21
- 트랜스포머 모델 입력크기(input-size)
  - batch_size: 32, sequence_length: 128, feature : 5



Single Head Attention : Query, Key, Value



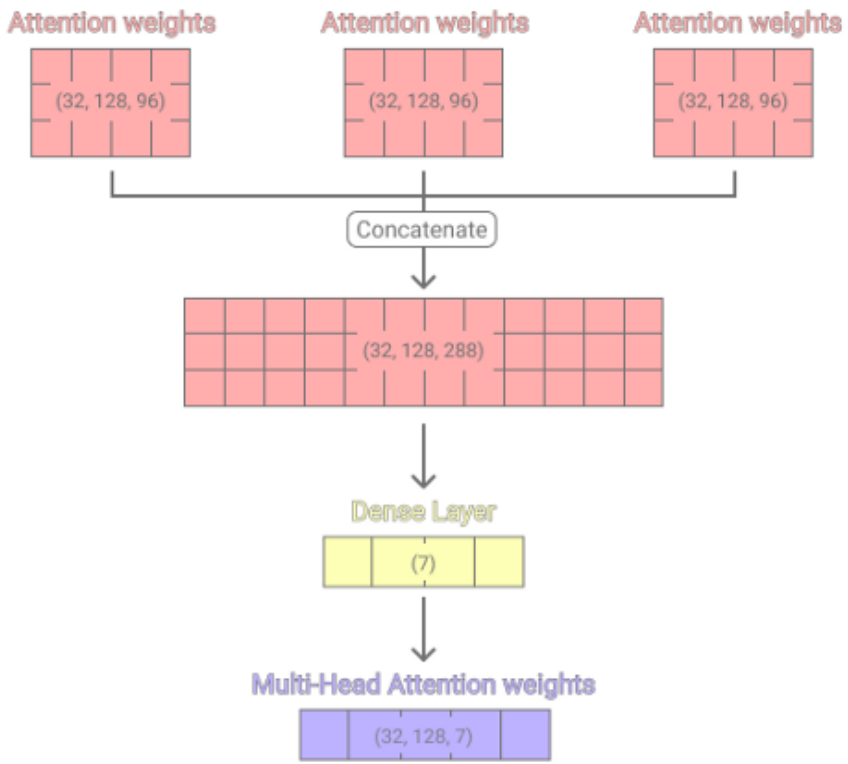
- Attention Weight는 Softmax를 사용함. Q*K*V



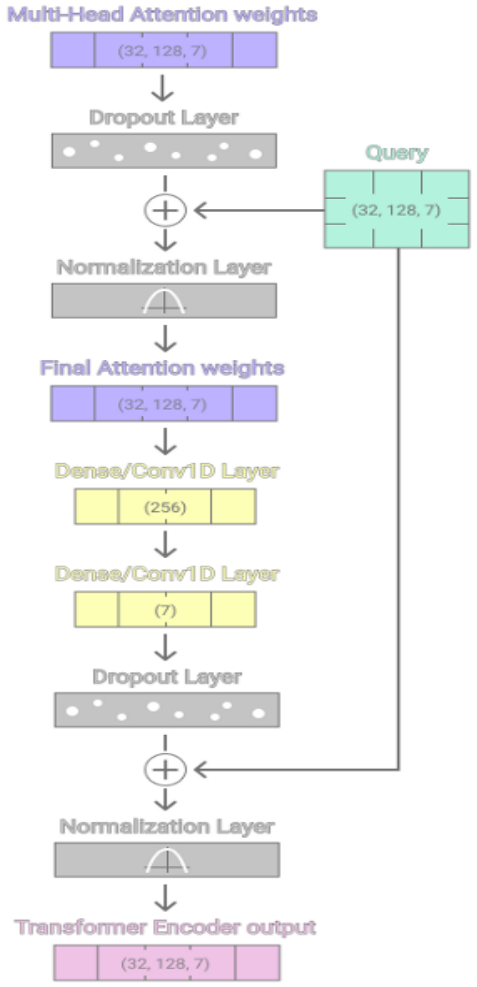
Transformer Traffic Model

Mulit-Head Attention으로 병렬처리가 가능함

❖ VDS 17, VDS 18 데이터 구조 (train, test, validation)

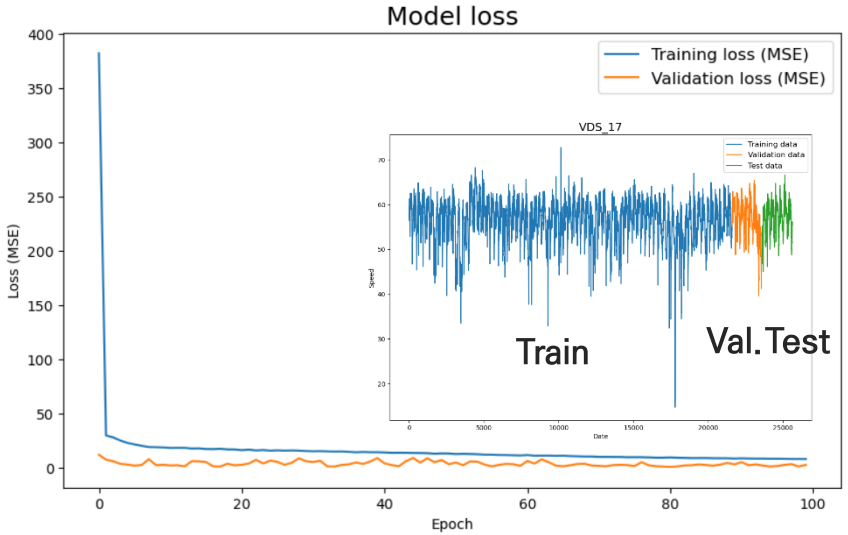


Transformer Encoder 출력이되 Decoder에 사용됨



Long-Term prediction of Traffic Speed and Volume

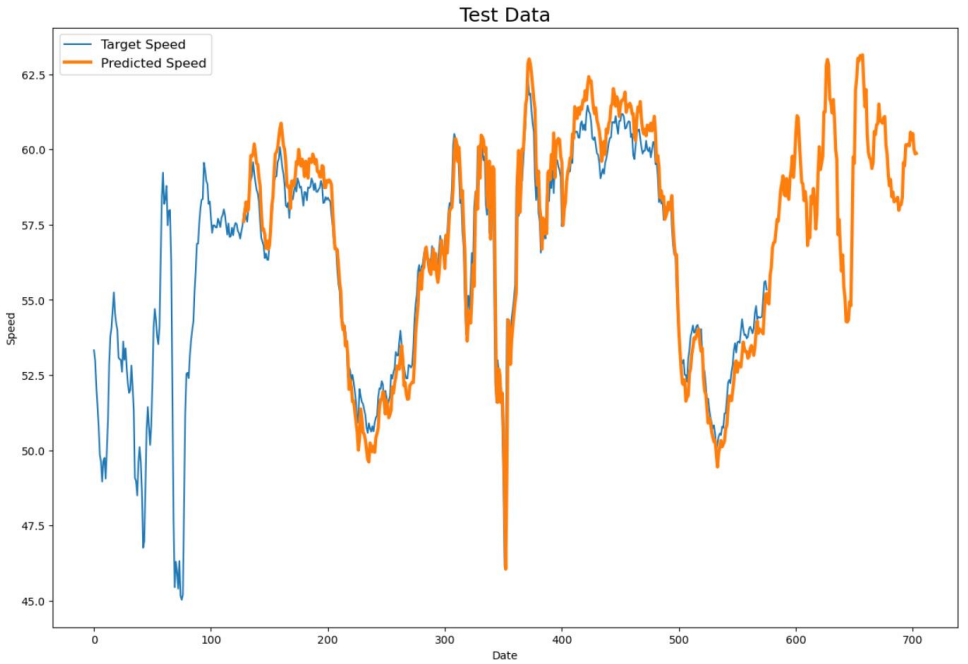
대전시 차량검지기(VDS) 데이터 105개



- 대학로 관련 5개 위치 데이터 90일
  - 데이터 개수: 25,920개
  - 90일*24*12 (5분 단위)

장기예측 성능 : 2016개 예측(1주일, 288*7)

- TransformerVDS: Transformer+TimeEmbedding

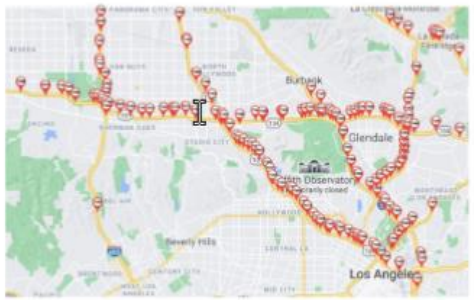


- TransformerVDS 결과와 대전 UVDS 데이터에 적용하고 있으며, SCI 논문에 투고할 예정

Dynamic Spatial Transformer WaveNet Network

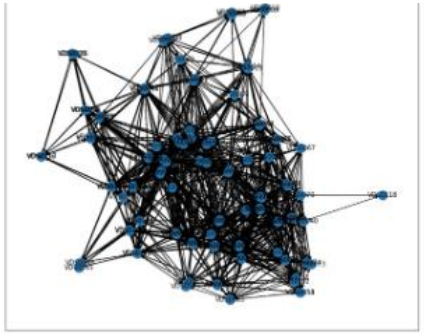
UVDS: 대전 VDS 데이터 기반 시공간 그래프 신경망 구축
Spatial-temporal Graph neural Network

- ❖ Metr-LA : 대표적 교통 공개 데이터 (미국, LA지역 4개월)
- ❖ UVDS 공개 : (시공간) 대전시 차량검지기 데이터 (3개월)

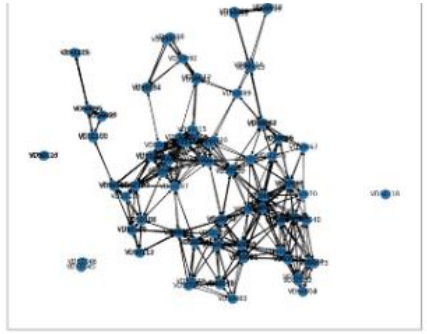


a) METR-LA

b) UVDS



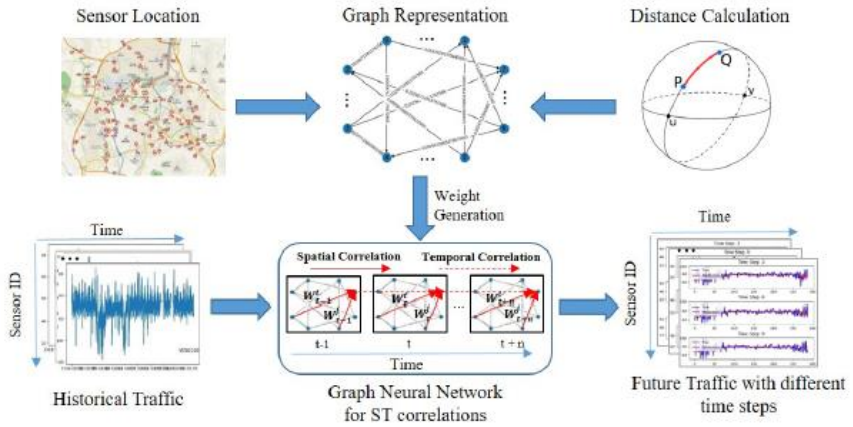
a) β = 0.1



b) β = 0.5

대전시 VDS 데이터 기반 새로운 UVDS 데이터 공개
UVDS: A New Dataset for Trac Forecasting with Spatial-Temporal Correlation

- ❖ General Framework for Traffic Forecasting using Spatial-temporal Correlaitons



Dynamic Spatial Transformer WaveNet Network

UVDS 데이터 기반 동적 시공간-트랜스포머 웨이브넷 신경망 개발 및 UVDS 데이터 성능 테스트
DSTWN 모델 : Dynamic Spatial Transformer WaveNet Network로 Spatial-Temporal Graph 신경망 보다 우수성 검증

❖ 기존 Spatial-Temporal Graph Network과 비교 성능 향상 목표

❖ 개발한 DSTWN 알고리즘 성능 벤치마크 : Metr-LA 데이터

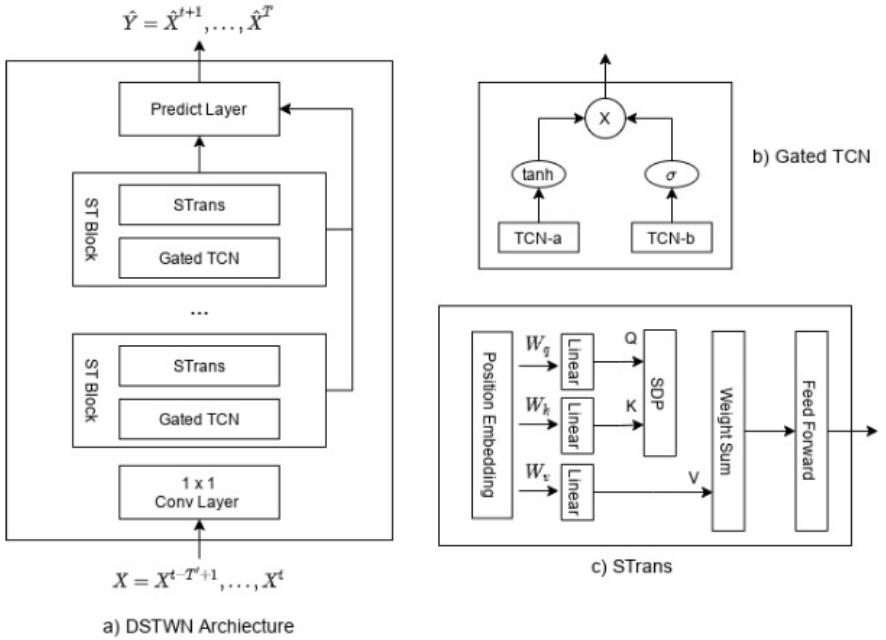


Fig. 3. The DSTWN Architecture

Table 2. Results on METR-LA dataset.

Step	Metrics	GraphWN	MTGNN	STWaNet	STTN	DSTWN
1	MAE	2.2408	2.2441	2.2791	2.4080	2.3220
	RMSE	3.8636	3.9185	3.9793	4.2339	4.0704
	MAPE	0.0540	0.0551	0.0550	0.0588	0.0567
3	MAE	2.7127	2.6762	2.7457	2.9160	2.8758
	RMSE	5.1690	5.1428	5.3162	5.6556	5.6285
	MAPE	0.0695	0.0686	0.0711	0.0778	0.0759
6	MAE	3.0974	3.0605	3.0947	3.3819	3.2909
	RMSE	6.1839	6.2002	6.2781	6.8651	6.7740
	MAPE	0.0847	0.0816	0.0838	0.0958	0.0900
9	MAE	3.3617	3.3100	3.3239	3.6961	3.5461
	RMSE	6.8279	6.8380	6.8639	7.5749	7.3837
	MAPE	0.0950	0.0912	0.0924	0.1085	0.0984
12	MAE	3.5760	3.4937	3.5036	3.9533	3.7446
	RMSE	7.2883	7.2421	7.2761	8.1262	7.8246
	MAPE	0.1035	0.0982	0.0993	0.1181	0.1047
Training		45.6927	62.8770	54.5744	77.5496	133.2374
Inference		1.4630	1.6825	1.5830	6.6945	3.8088

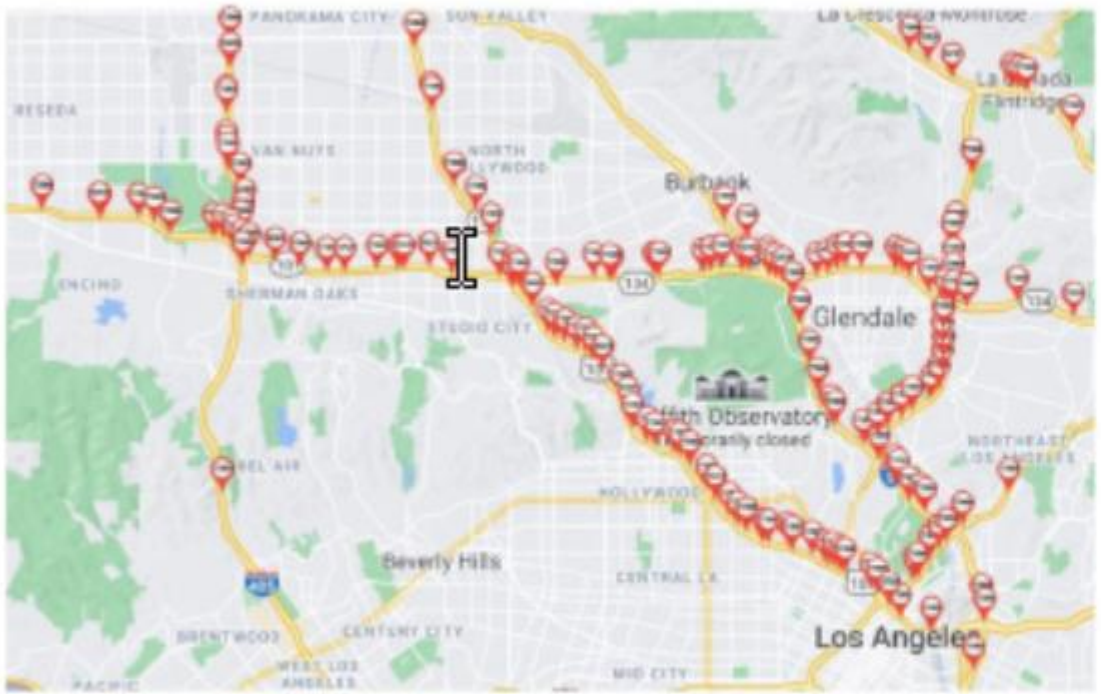
Public spatial–temporal tra±c datasets

Table 1. Public datasets for spatial–temporal traffic prediction task.

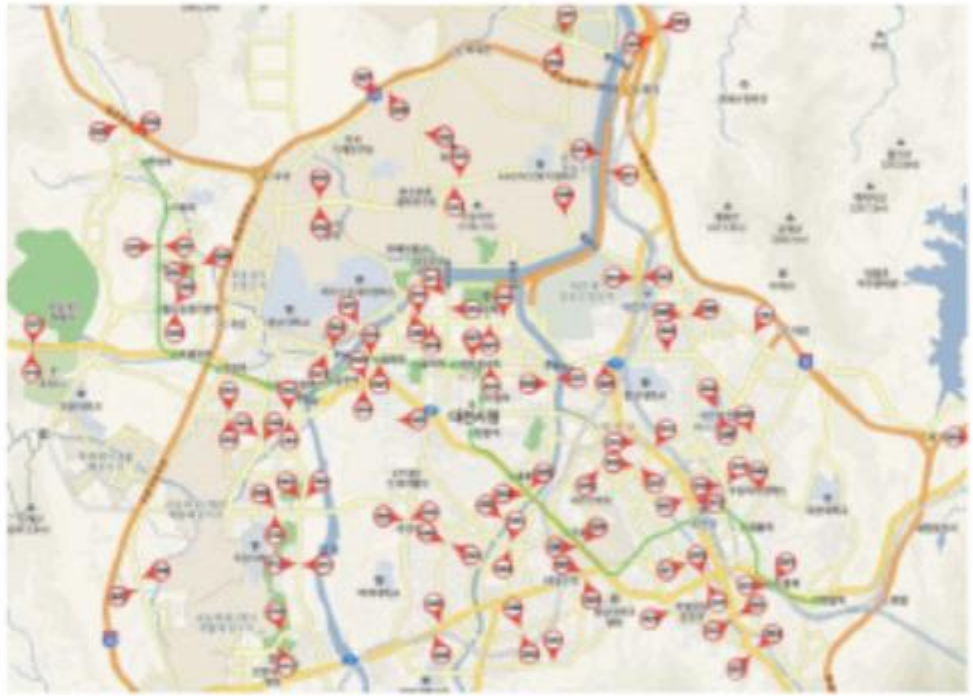
Datasets	Characteristic	Time interval (min)	Source (https://github.com)
PeMS-BAY	Speed	5	/liyaguang/DCRNN (2018)
METR-LA	Speed	5	/liyaguang/DCRNN (2018)
LOOP	Speed	5	/zhiyongc/Seattle-Loop-Data (2018)
PeMSD4,8	Volume	5	/Davidham3/ASTGCN (2018)
Q-Traffic	Speed	15	/JingqingZ/BaiduTraffic (2018)
NYC Taxi	Demand	30	/toddwschneider/nyc-taxi-data (2019)
PeMSD3,7	Volume	5	/Davidham3/STSGCN (2020)

UVDS: Urban VDS Dataset

- Tra±c data have been collected from VDS
  - ✓speed, traffic volume, occupancy, and vehicle types.



(a) METR-LA



(b) UVDS

- For graph construction, UVDS based on the geometric distances between sensors (N=104)

$$W_{i,j} = \begin{cases} \exp\left(-\frac{d^2_{v_i,v_j}}{\sigma^2}\right), & \text{if } \exp\left(-\frac{d^2_{v_i,v_j}}{\sigma^2}\right) \geq \beta, \\ 0, & \text{otherwise,} \end{cases}$$

$$d_{v_i,v_j} = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\phi_j - \phi_i}{2}\right) + \cos(\phi_i)\cos(\phi_j)\sin^2\left(\frac{\varphi_j - \varphi_i}{2}\right)}\right)$$

Results on UVDS dataset

- Bolded texts are best results

Step	Metrics	MTGNN	STAWnet	DSTWN
1	MAE	3.4671	3.4535	**3.4501**
	RMSE	5.2665	**5.2494**	5.2517
	MAPE	0.0746	0.0740	**0.0736**
3	MAE	**3.6504**	3.6609	3.6577
	RMSE	**5.5896**	5.6015	5.6105
	MAPE	0.0800	0.0804	**0.0799**
6	MAE	**3.7740**	3.8000	3.7903
	RMSE	**5.8132**	5.8491	5.8417
	MAPE	**0.0832**	0.0846	0.0840
9	MAE	3.8629	3.8753	**3.8556**
	RMSE	5.9516	5.9681	**5.9507**
	MAPE	**0.0856**	0.0865	0.0859
12	MAE	3.9589	3.9469	**3.9144**
	RMSE	6.0799	6.0687	**6.0362**
	MAPE	0.0887	0.0886	**0.0872**
Training (s/epoch)		16.7419	23.1326	30.4803
Inference (s/epoch)		0.4241	0.5603	0.7769

OPEN ACCESS
Vietnam Journal of Computer Science
(2022)

DOI: 10.1142/S2196888822500324

Transformer Based Traffic flow prediction Research

KISTI SINCE 1962

- 대전시 데이터웨어하우스에서 차량검지기(VDS)와 RSE 데이터 수집 및 분석함
  - ✓KISTI 정문 앞 도로 차량검지기(VDS 17데이터)는 오전과 출근과 오후 퇴근에 일부 속도가 떨어지는 경향이 있음.
  - ✓대학로 인근 RSE 데이터는 전체적으로 속도가 너무 낮게 측정되었음.

- (딥러닝 기반 교통 흐름 예측)
  - ✓RNN 기반 LSTM을 양방향 장단기로 교통 흐름 예측하였고 교통 혼잡은 없는 것으로 예측됨
  - ✓Transformer_VDS 모델 개발로 장기 교통흐름 예측 정확도가 향상됨
  - ✓Spatial-Temporal Graph 데이터 기반 Dynamic Spatial Transformer WaveNet 모델 개발 및 성능 테스트 함

KISTI SINCE 1962

2022 TRUST KISTI

Korea Institute of Science and Technology Information