KiSTi 한국과학기술정보연구원 Korea Institute of Science and Technology Information

2022

Advanced Topic in Research Data-centric Deep Learning

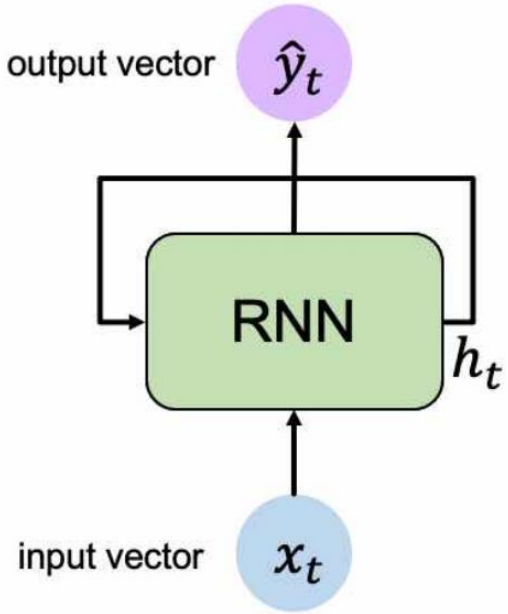
Lec 10: Seq2Seq and Attention

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

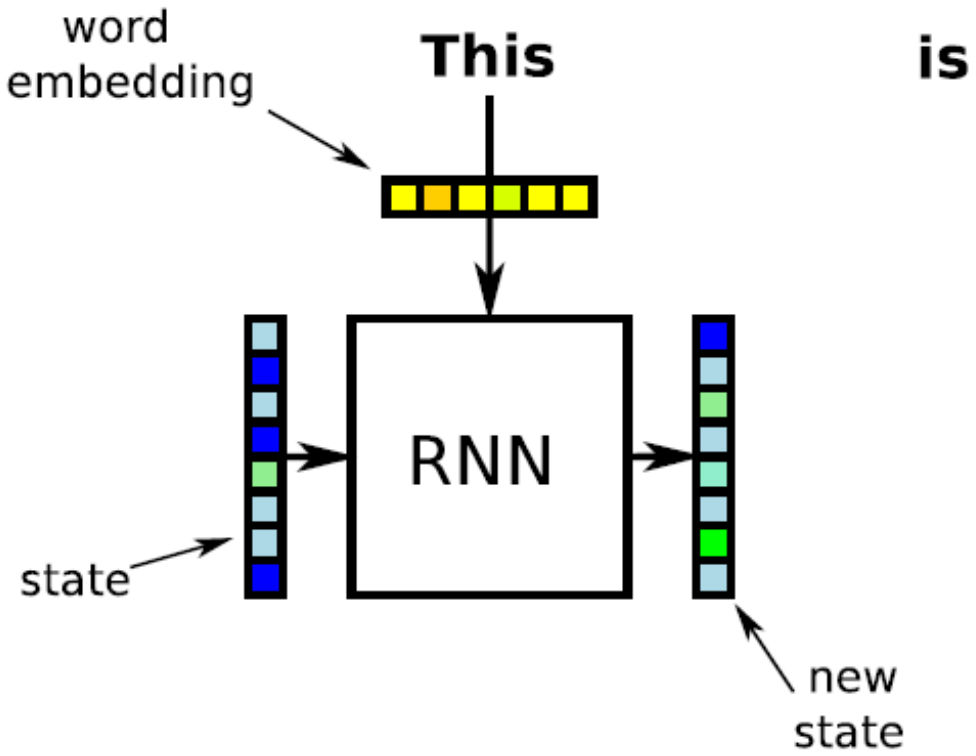


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

example: using the RNN output in a document classifier

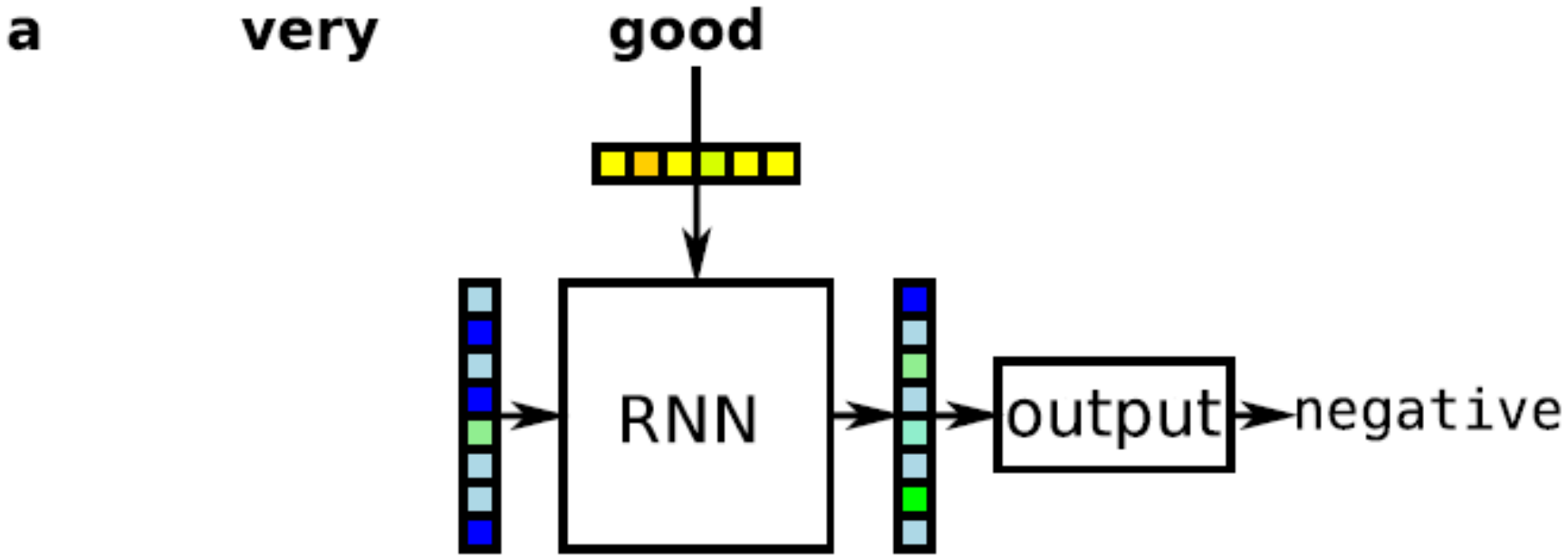


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

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

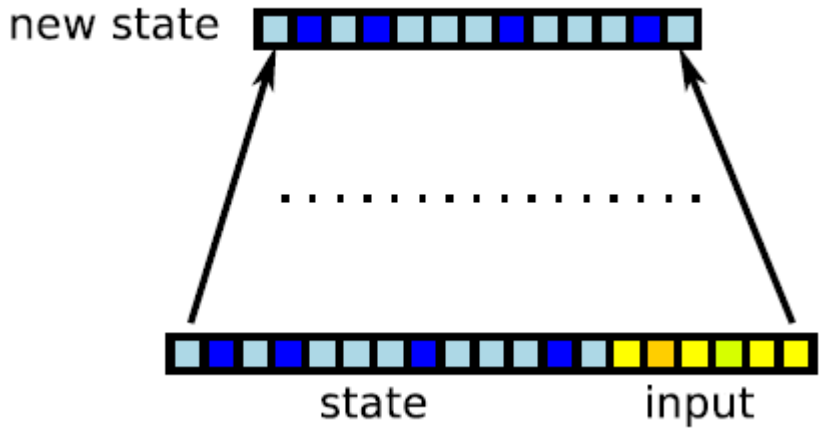


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

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

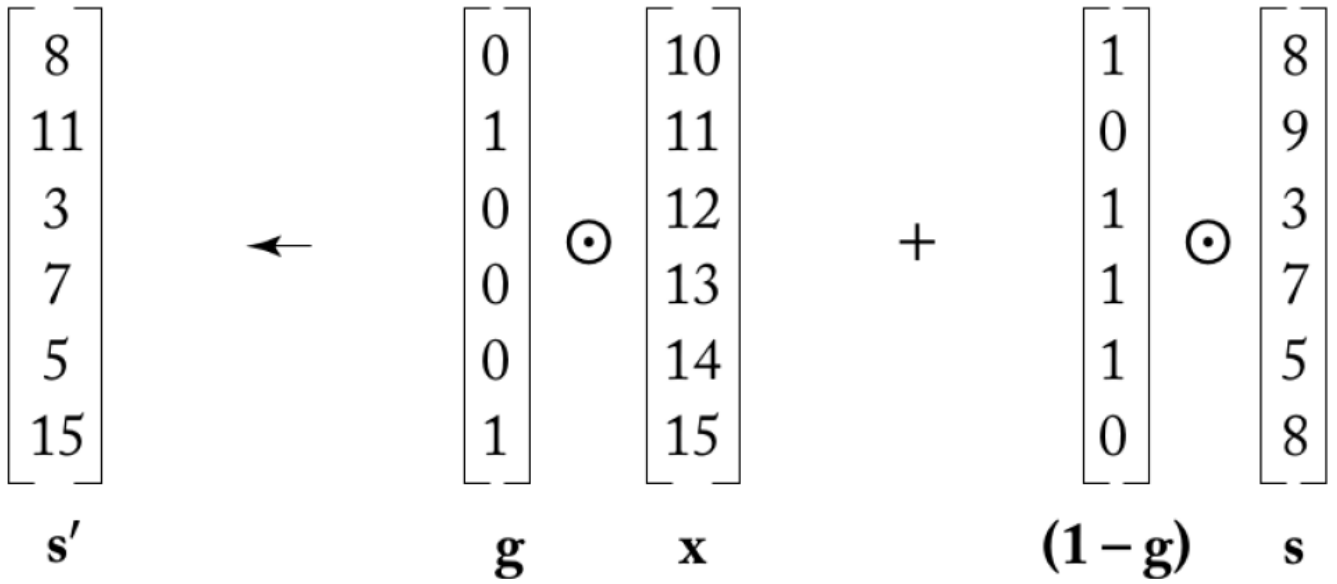


image from Goldberg's book

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

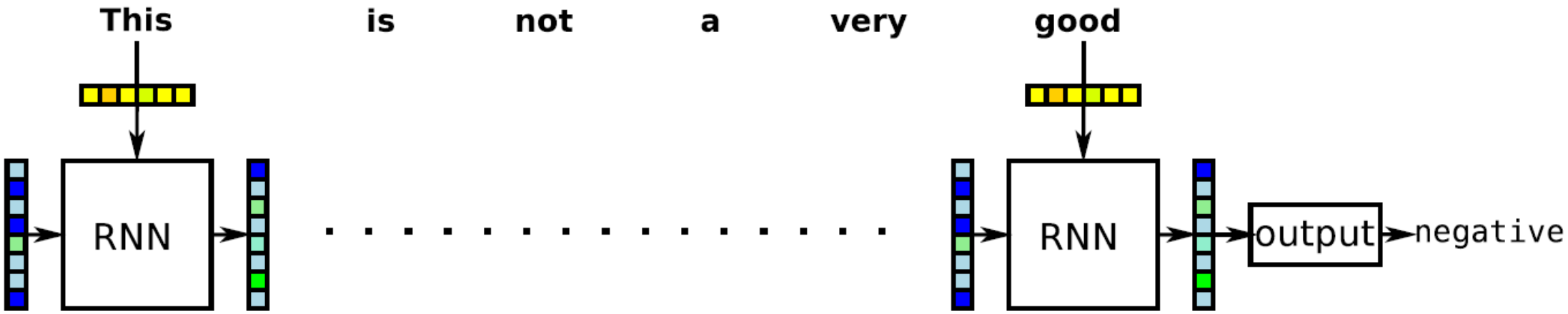


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

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)

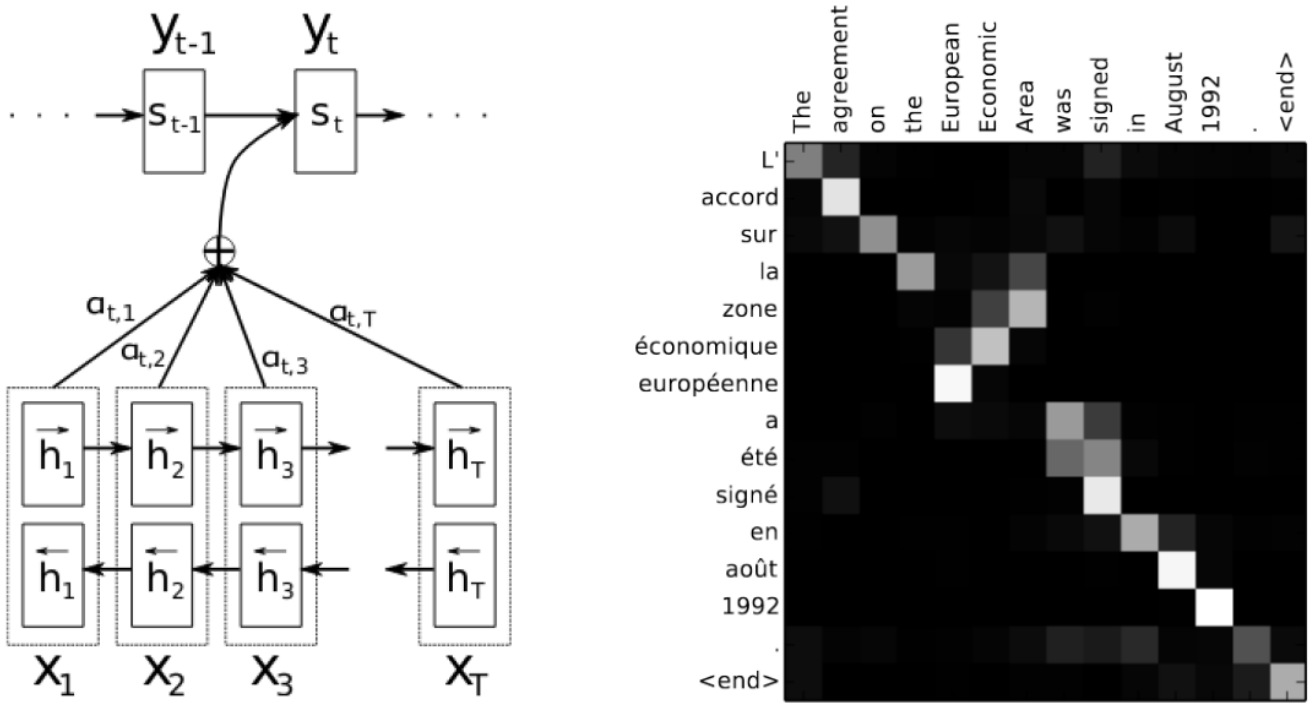


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

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"

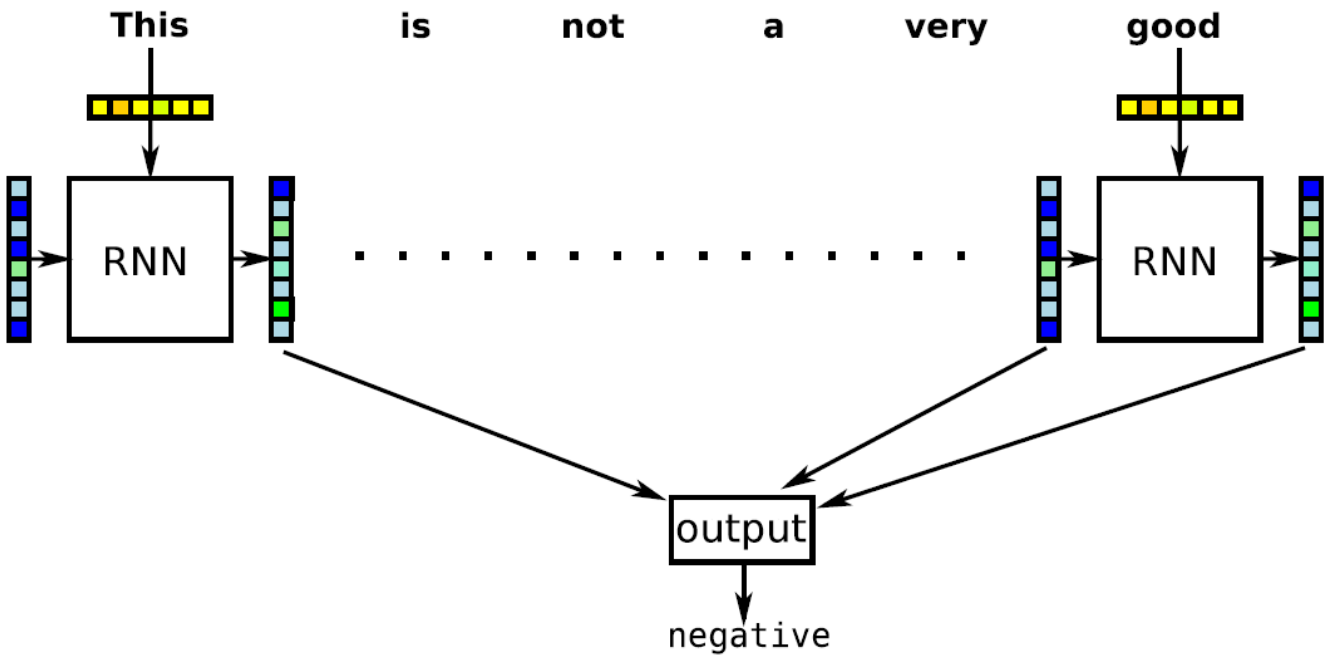


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Attention: a general formulation

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

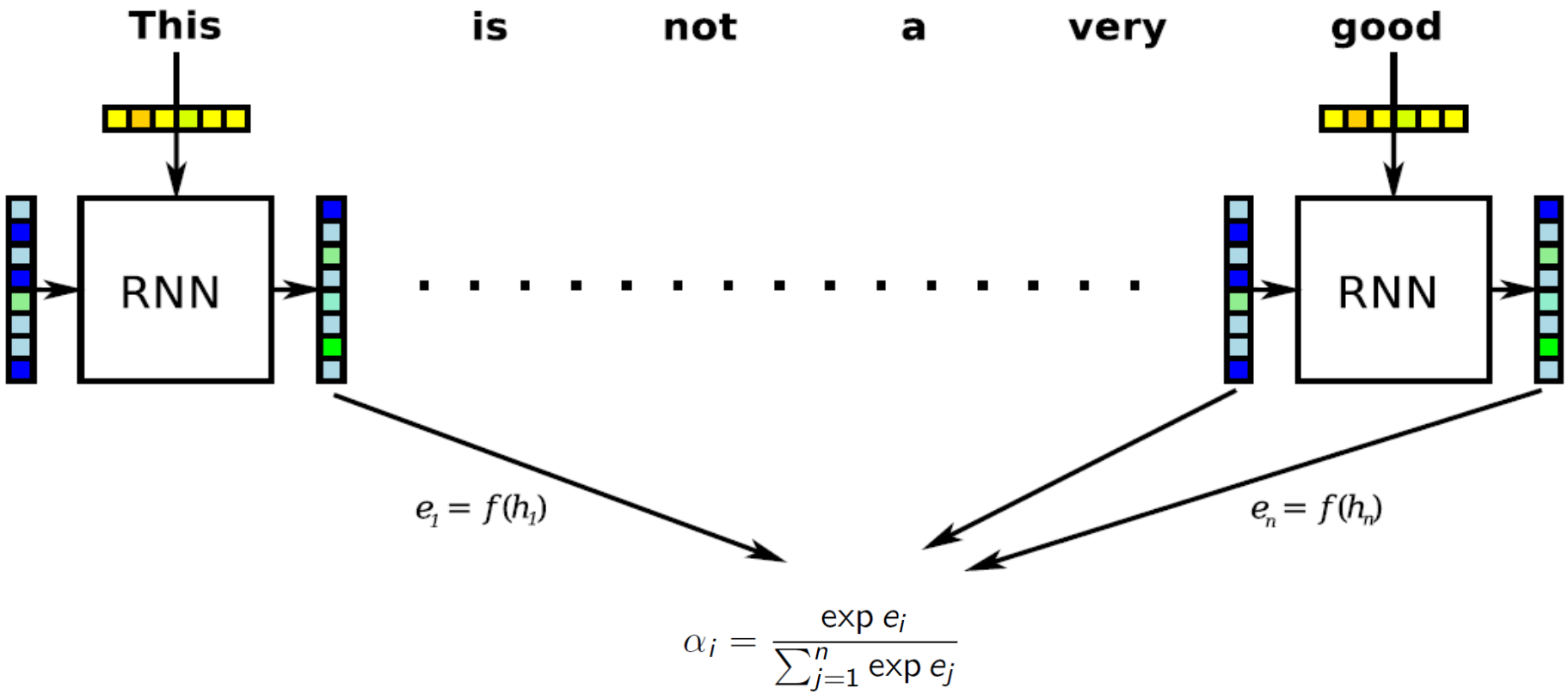


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Attention: a general formulation

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

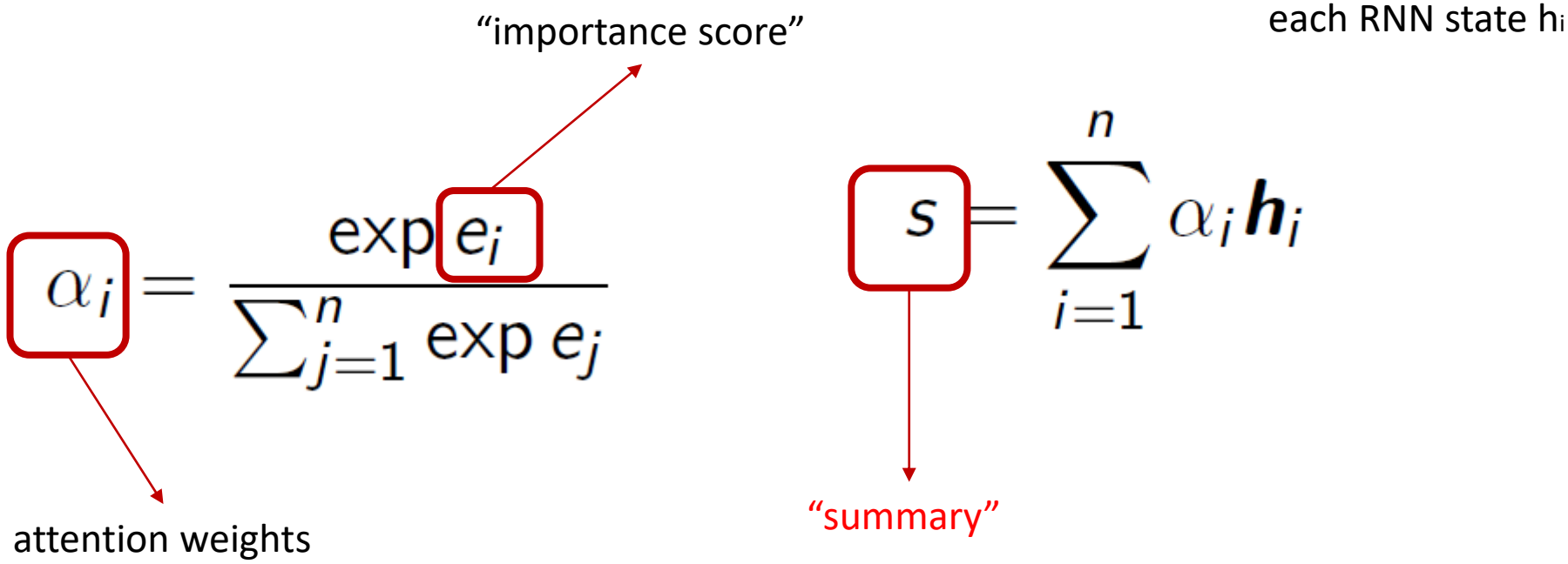


image borrowed from Richard Johansson (*Chalmers Technical University and University of Gothenburg)*

Neural Machine Translation

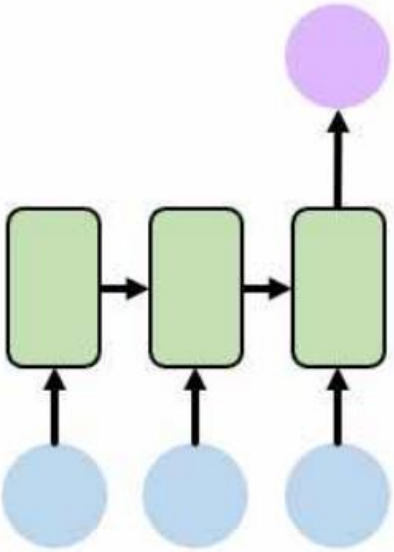
Sequence Modeling Applications

One to One

Binary Classification

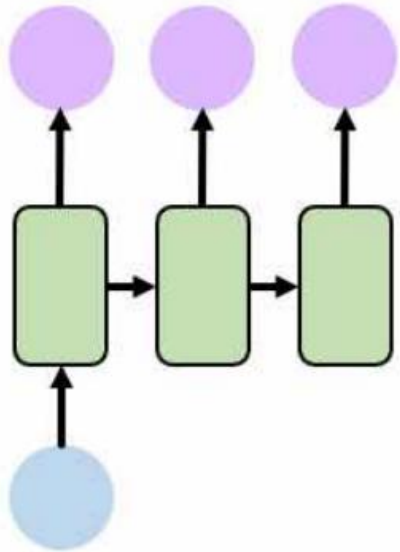
"Will I pass this class?"

Student → Pass?



Many to One

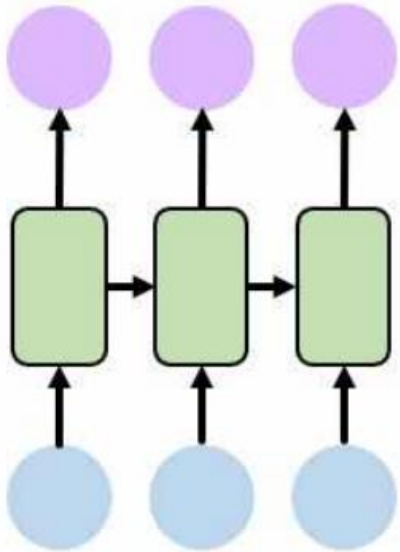
Sentiment Classification



One to Many

Image Captioning

"A baseball player throws a ball."



Many to Many

Machine Translation

What is a Seq2Seq model?

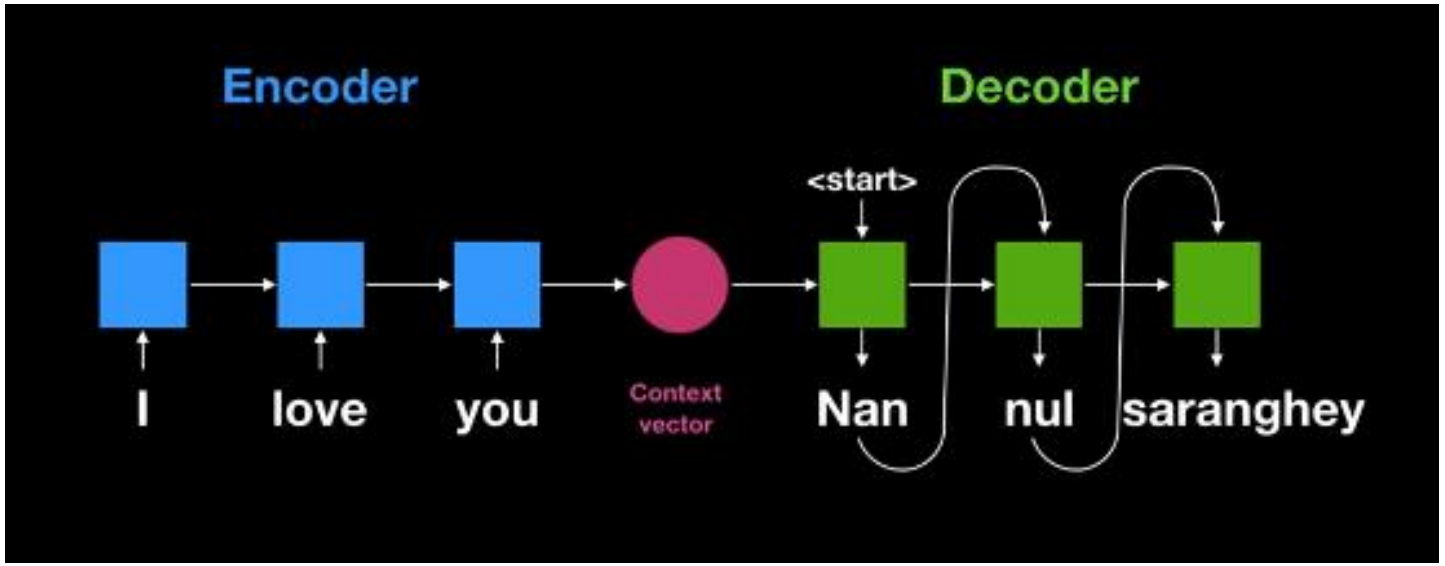
- Sequence-to-sequence learning (Seq2Seq)
  - ✓ Seq2Seq is about training models to convert sequences from one domain (e.g. sentences in English) to sequences in another domain (e.g. the same sentences translated to French).
- This can be used for machine translation

```
"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

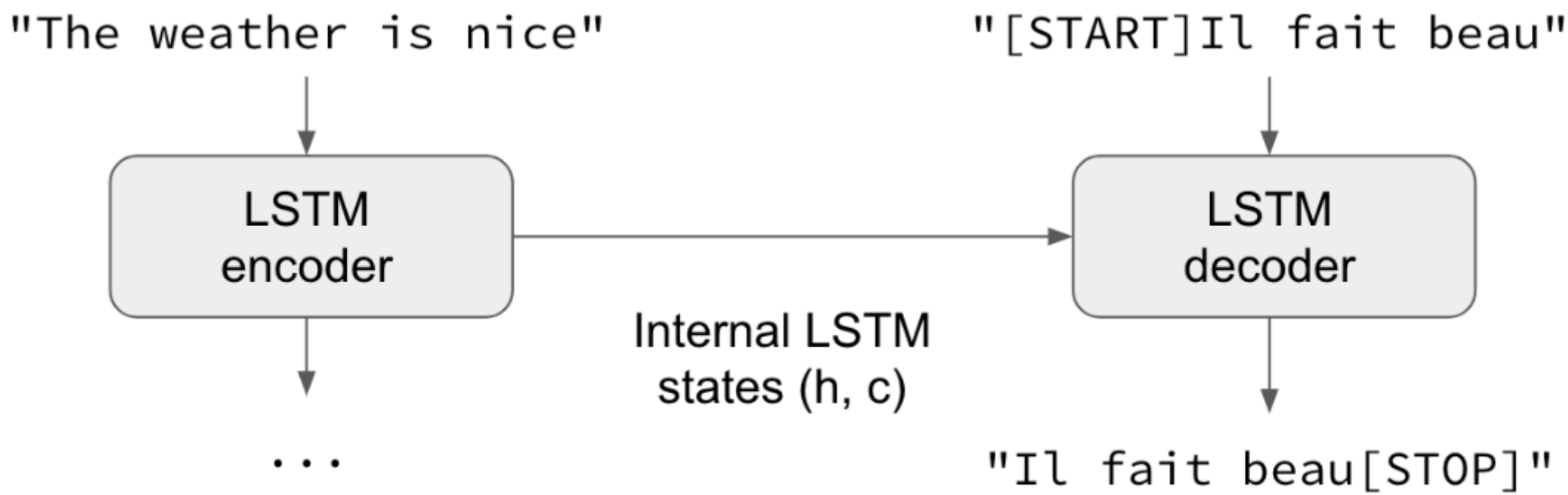
How does a classic Seq2Seq model work?

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

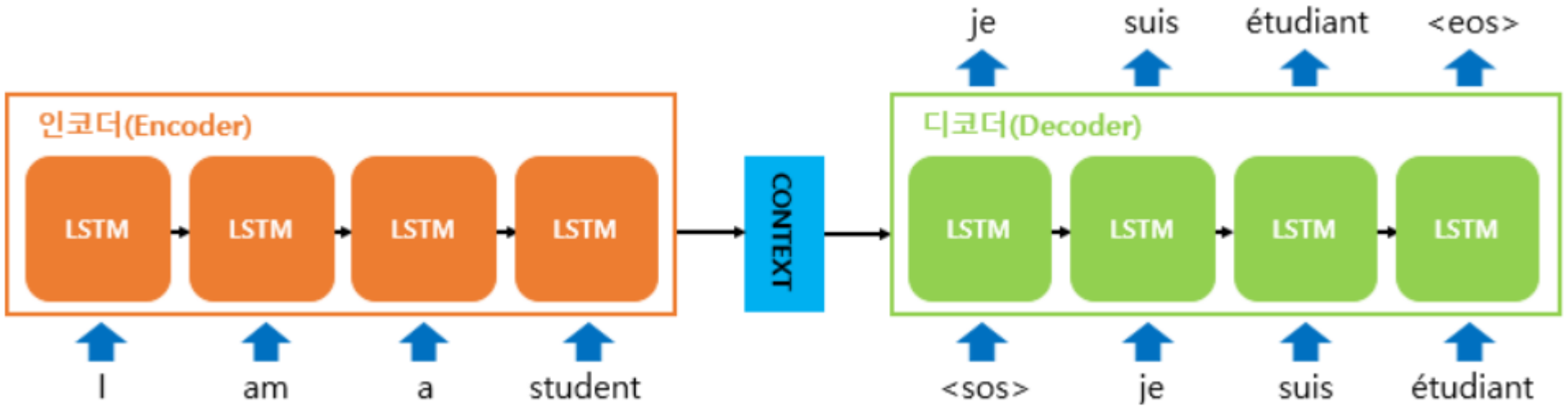
- Input sequences and output sequences have different lengths



Encoder and decoder are RNN architectures.

❖ context vector

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

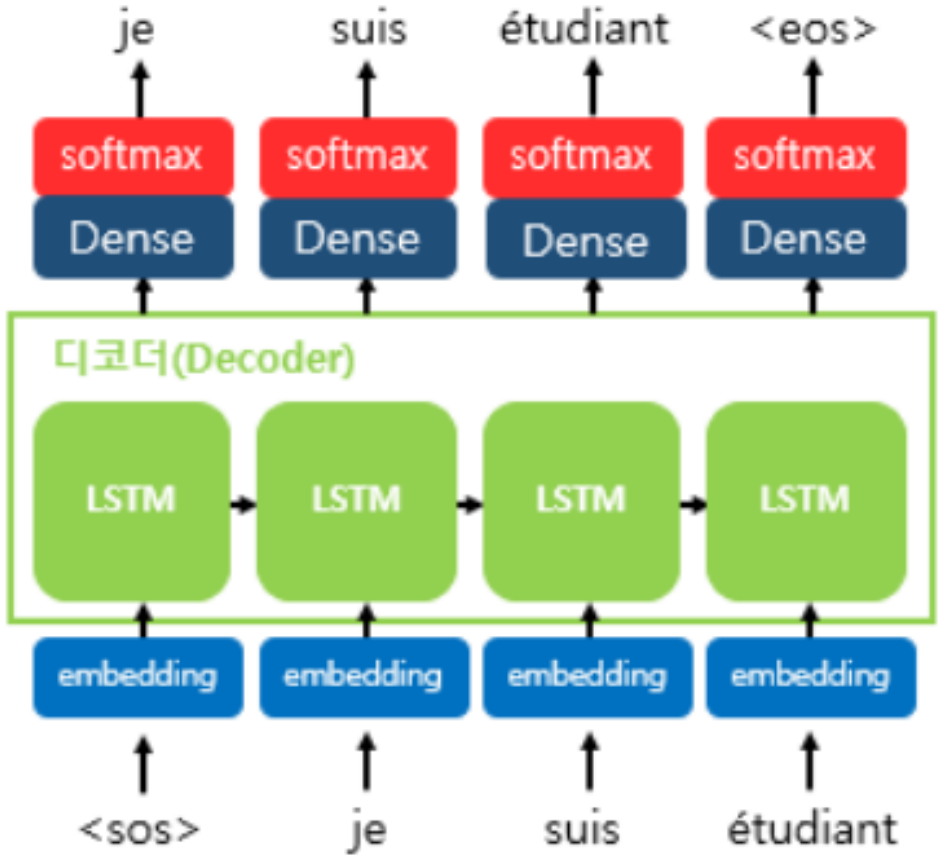


✓ Decoder is essentially an RNNLM (RNN Language Model)

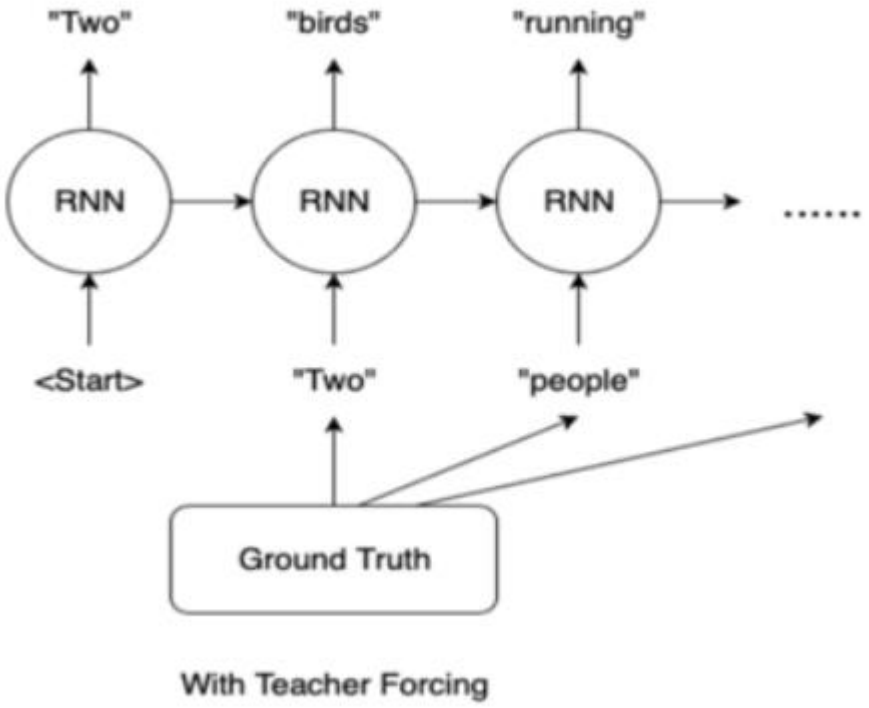
- Many-to-Many

Seq2Seq: Decoder part

- Softmax for next prediction word

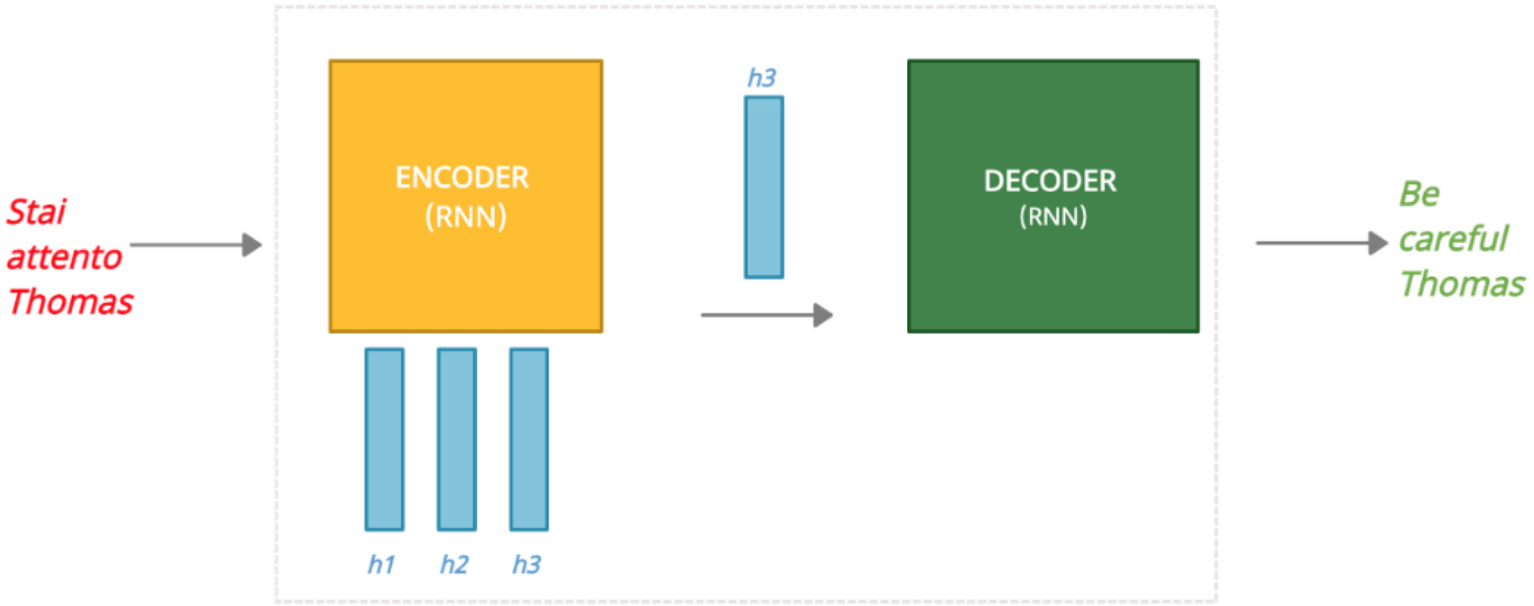


Teacher Forecing Learning



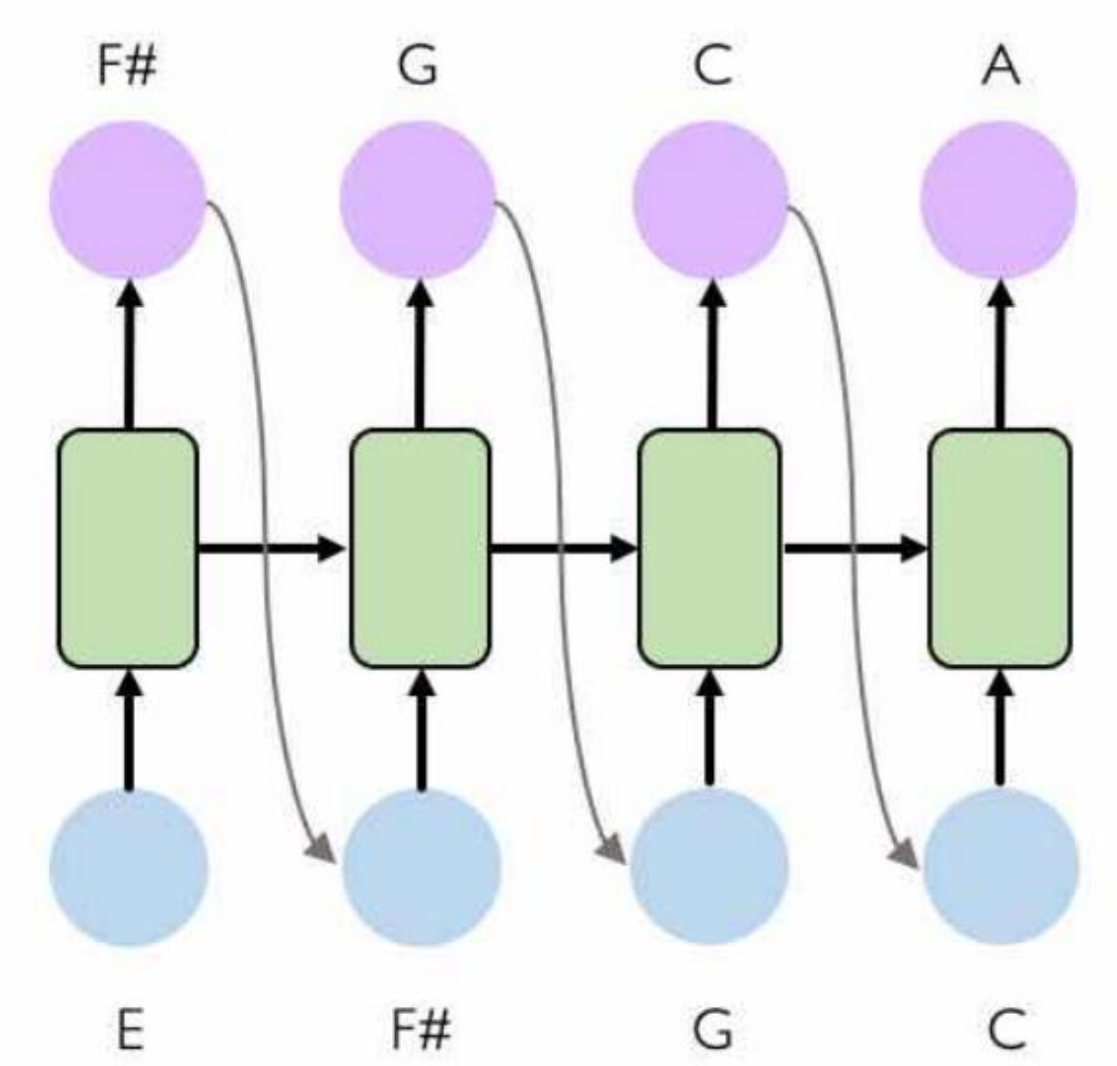
Limitation of Seq2Seq model

- 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

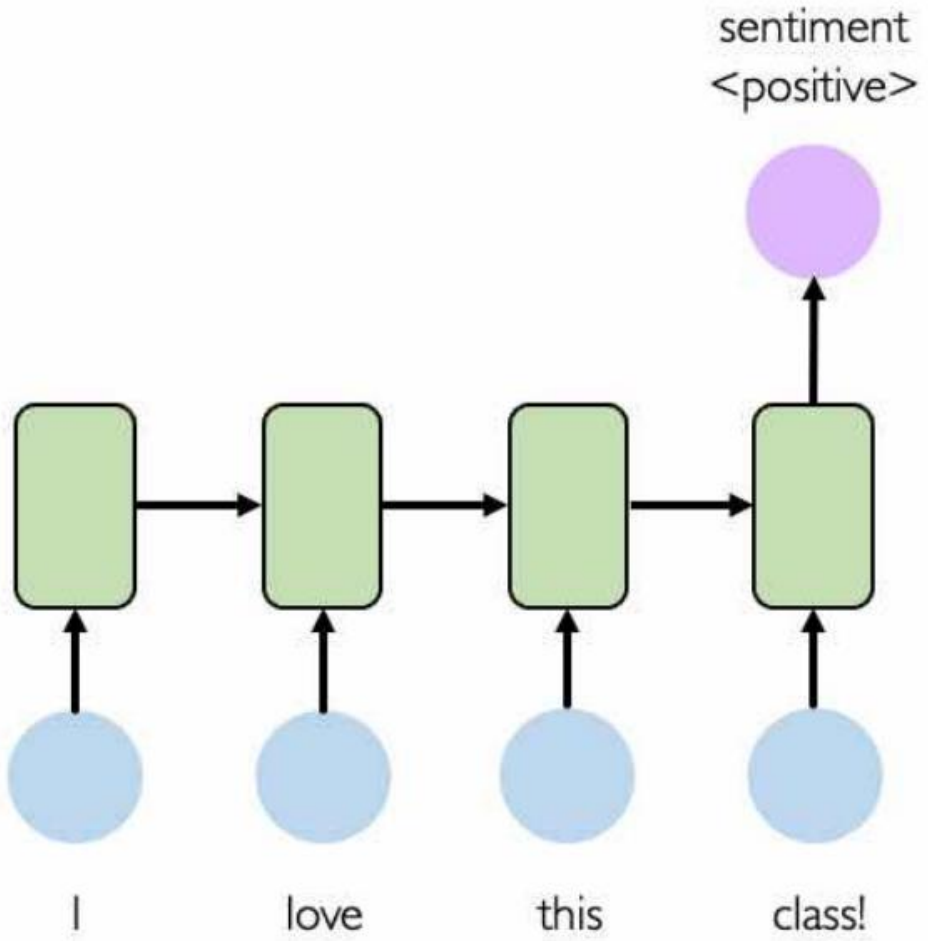


RNN Applications and Limitations

Example Task : Music Generation



Example Task : Sentiment Classification

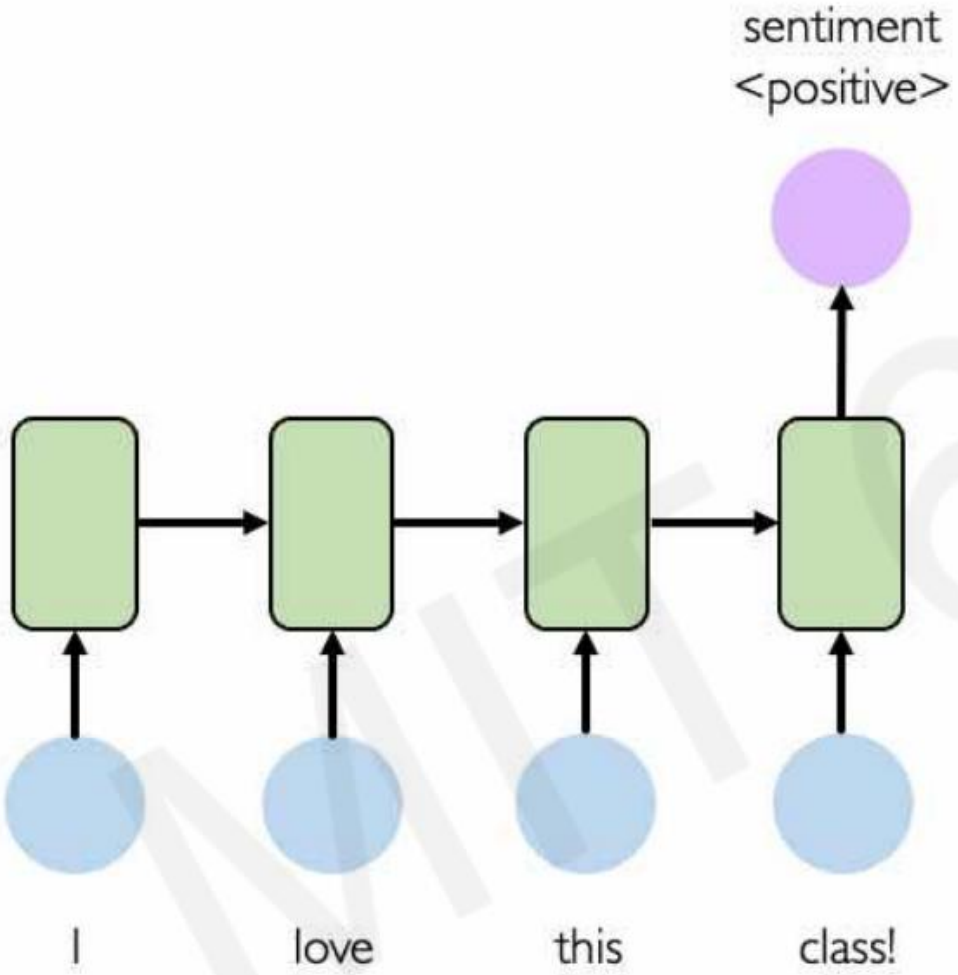


Input: sequence of words

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

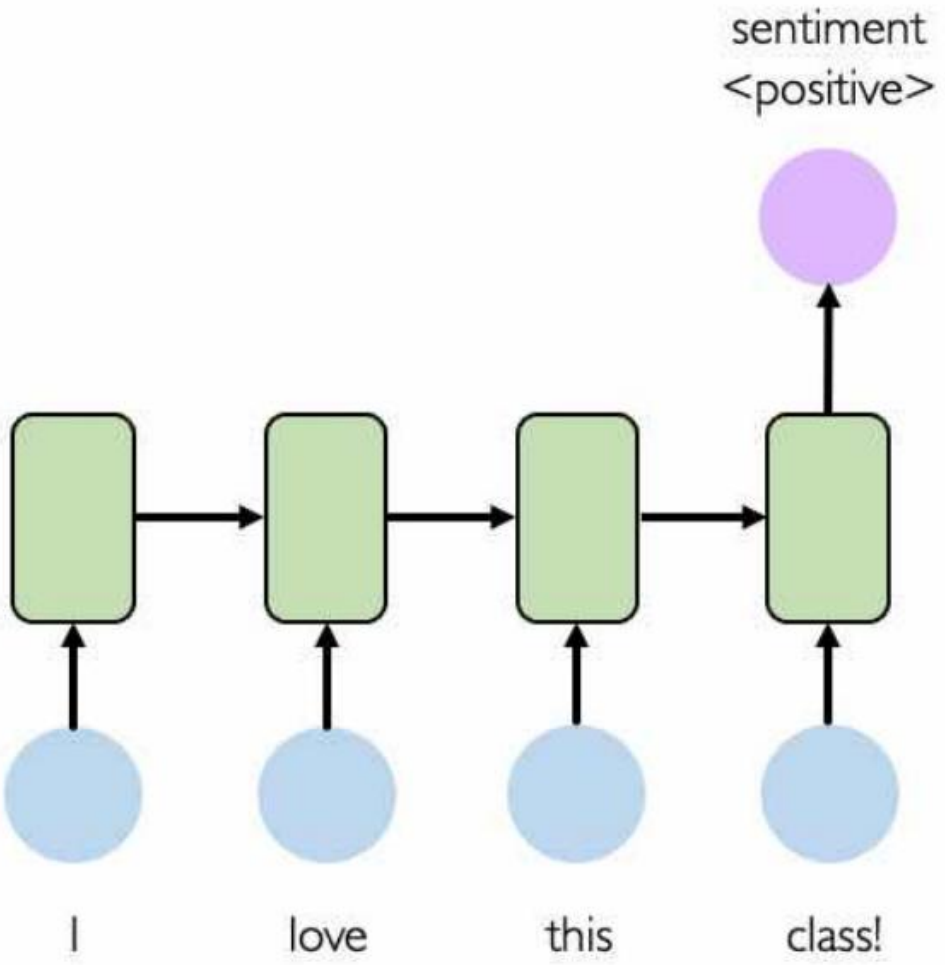
Angels-Cave
@AngelsCave
Follow

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

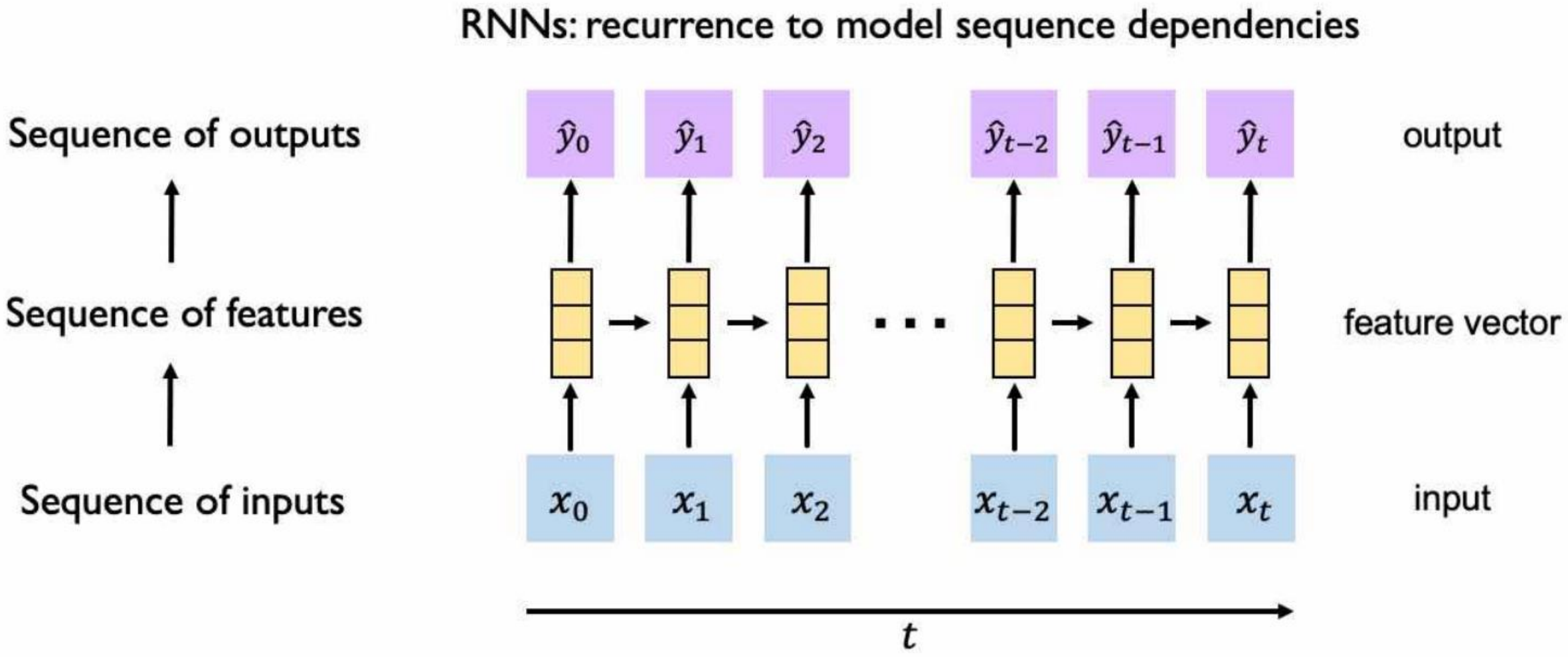
Limitations fo Recurrnet Models



Limitations of RNNs

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

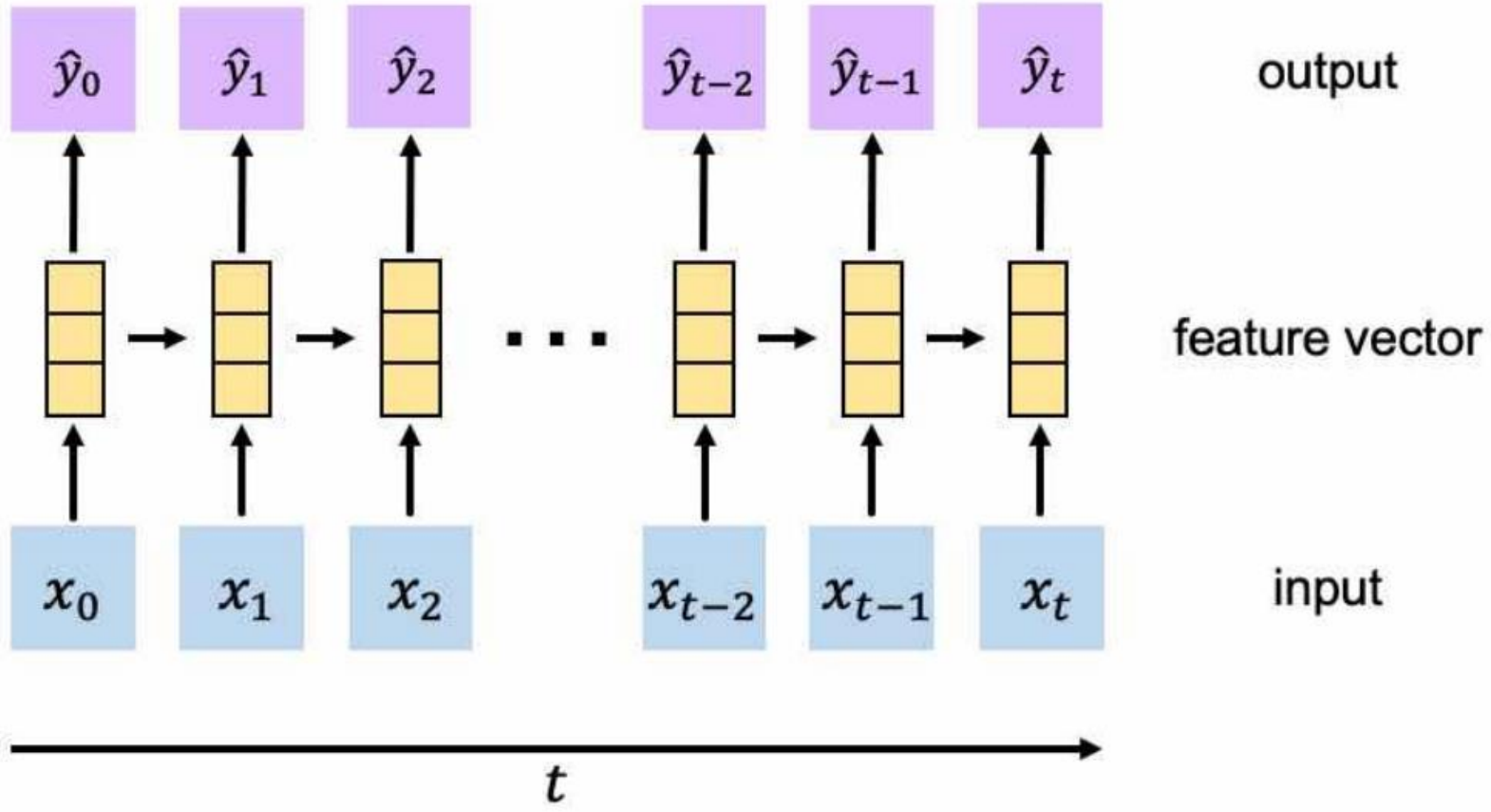
Goal of Sequence Modeling



Goal of Sequence Modeling

Limitations of RNNs

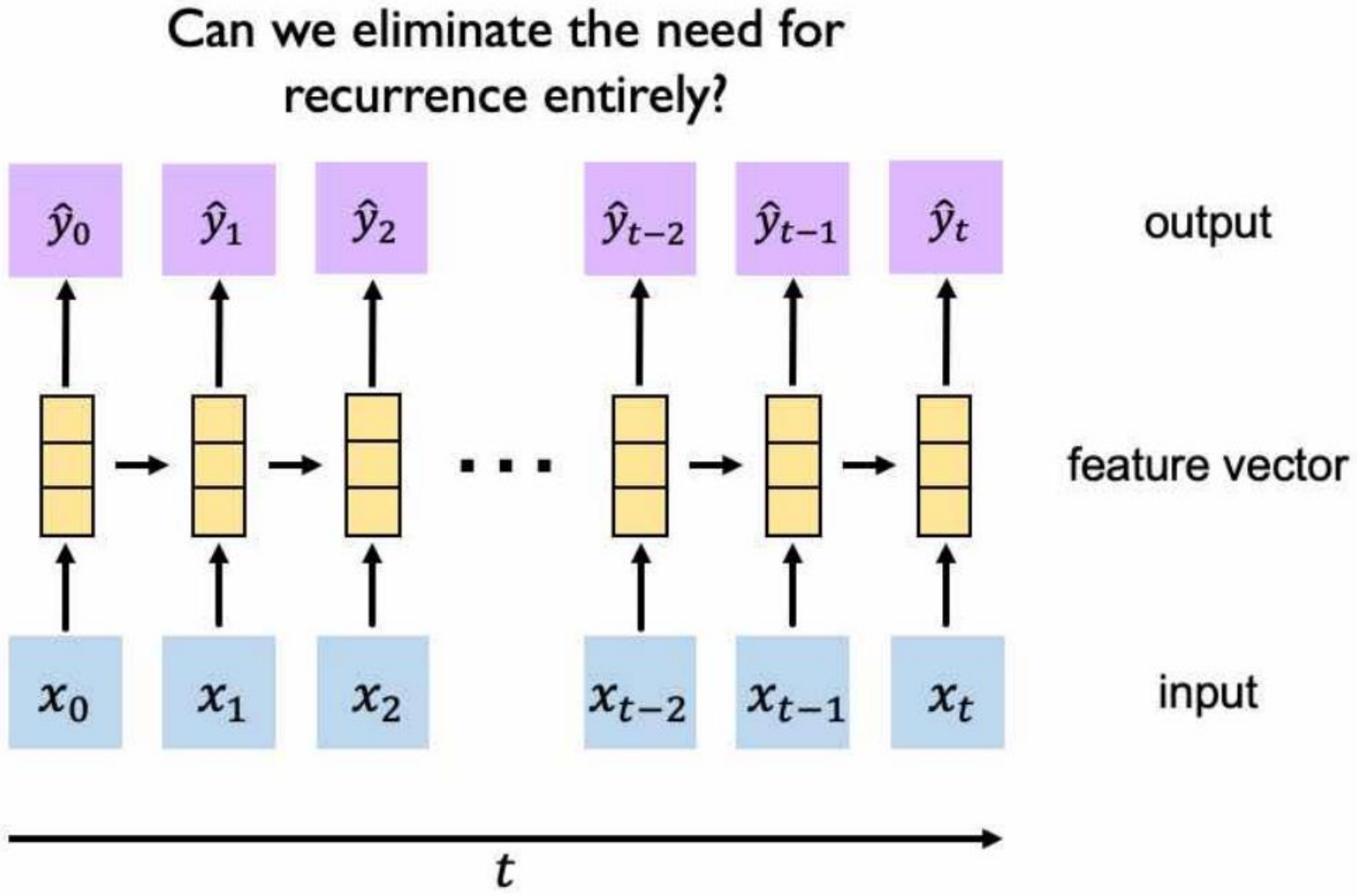
- Encoding bottleneck
- Slow, no parallelization
- Not long memory



Goal of Sequence Modeling

Desired Capabilities

- Continuous stream
- Parallelization
- Long memory

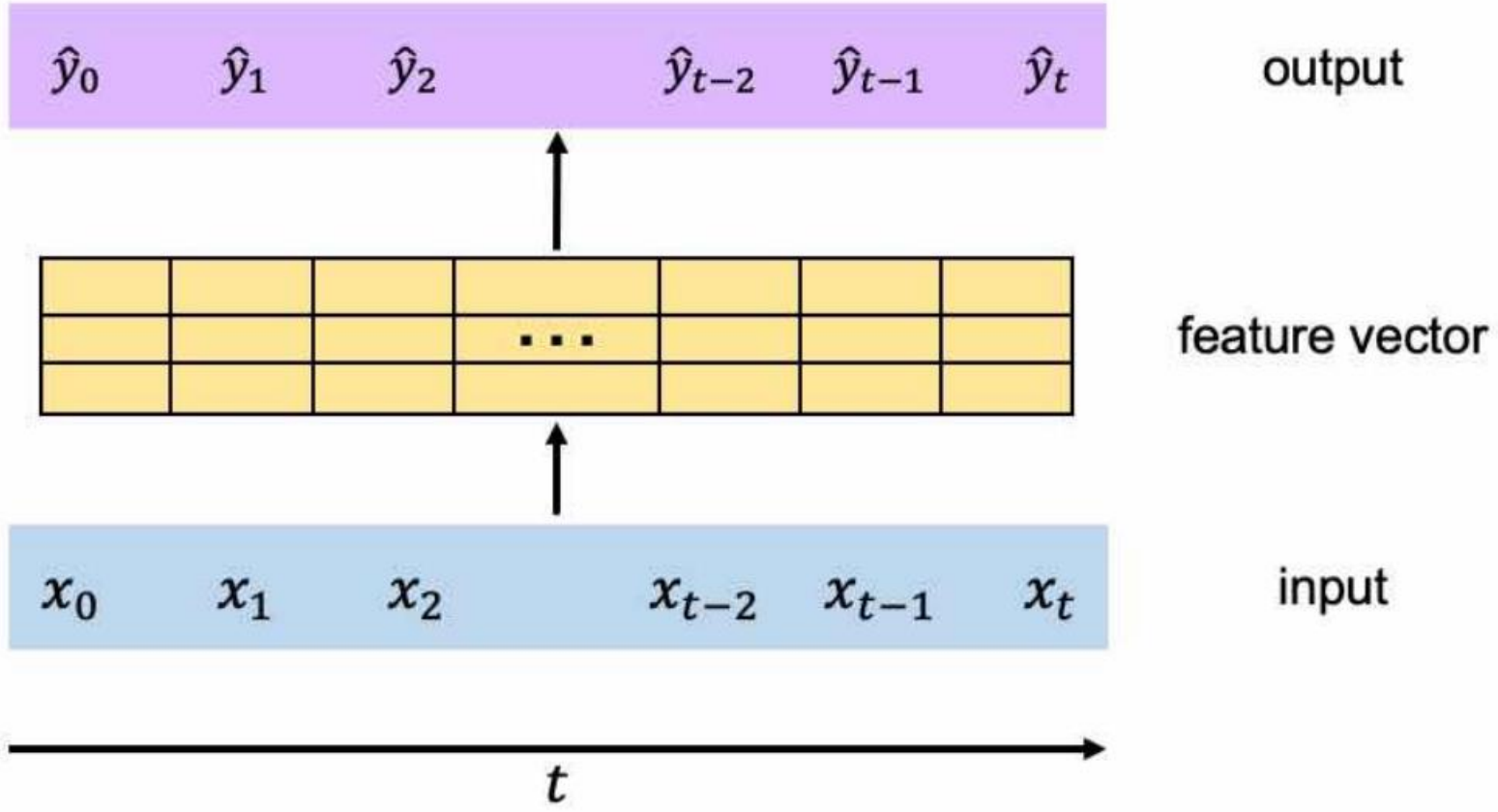


Goal of Sequence Modeling

Desired Capabilities

- Continuous stream
- Parallelization
- Long memory

Can we eliminate the need for recurrence entirely?



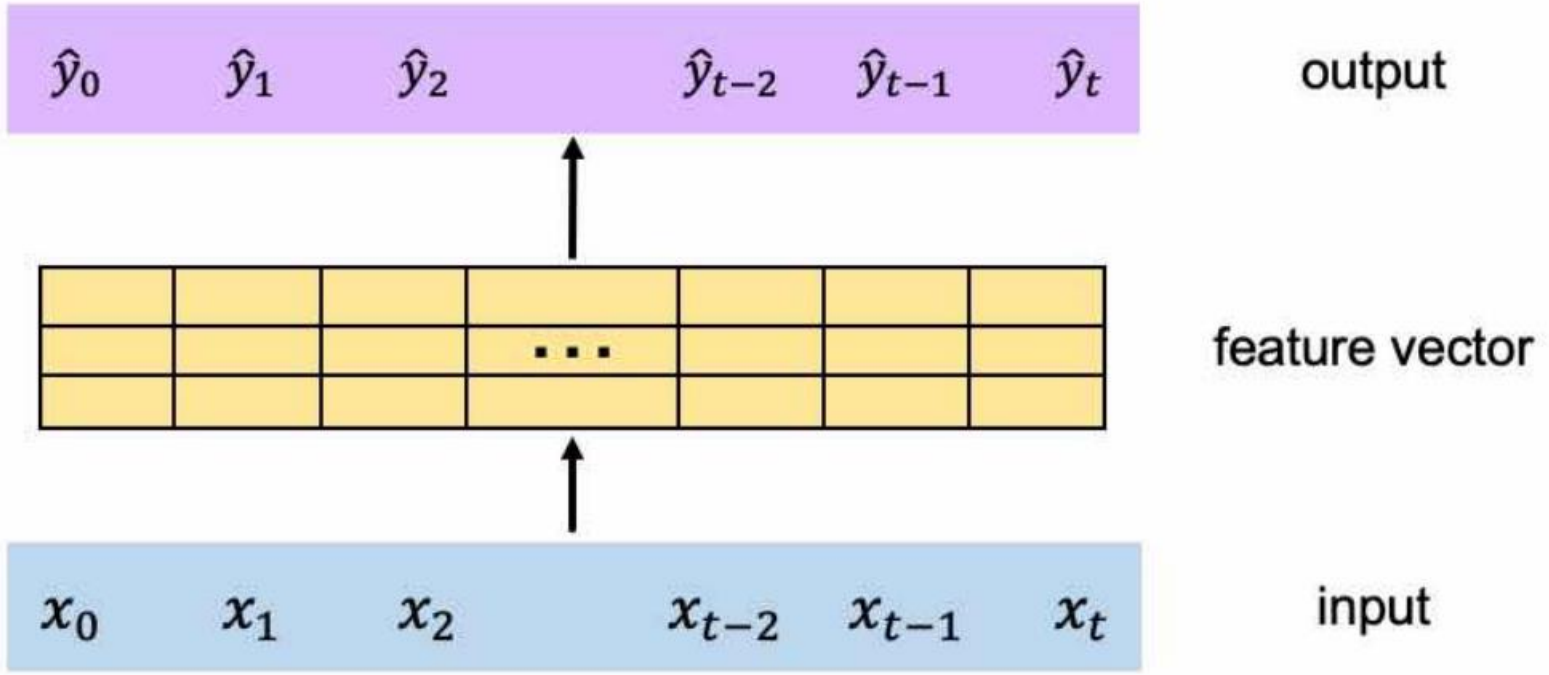
Goal of Sequence Modeling

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

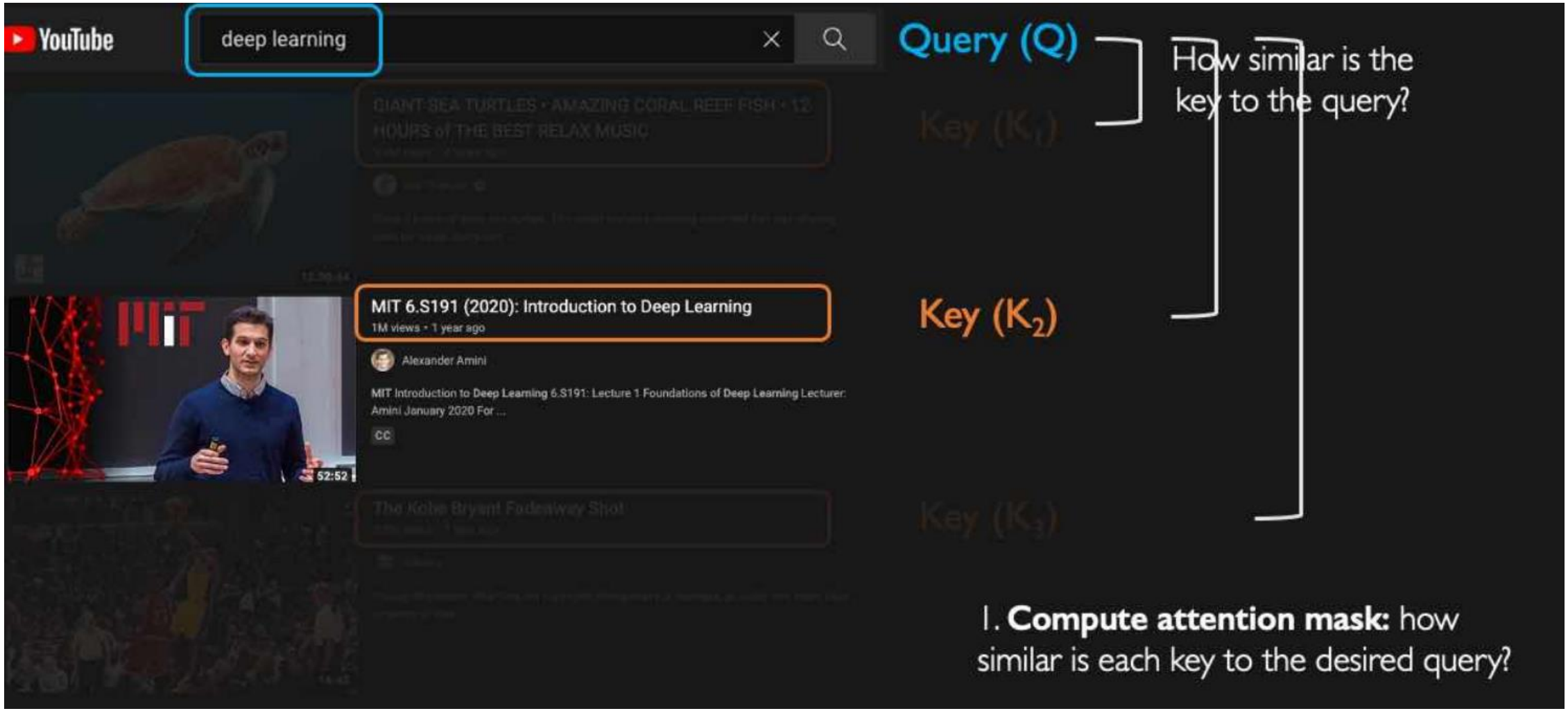
Intution Behind Self-Attention

Attending to the most important parts of an input.

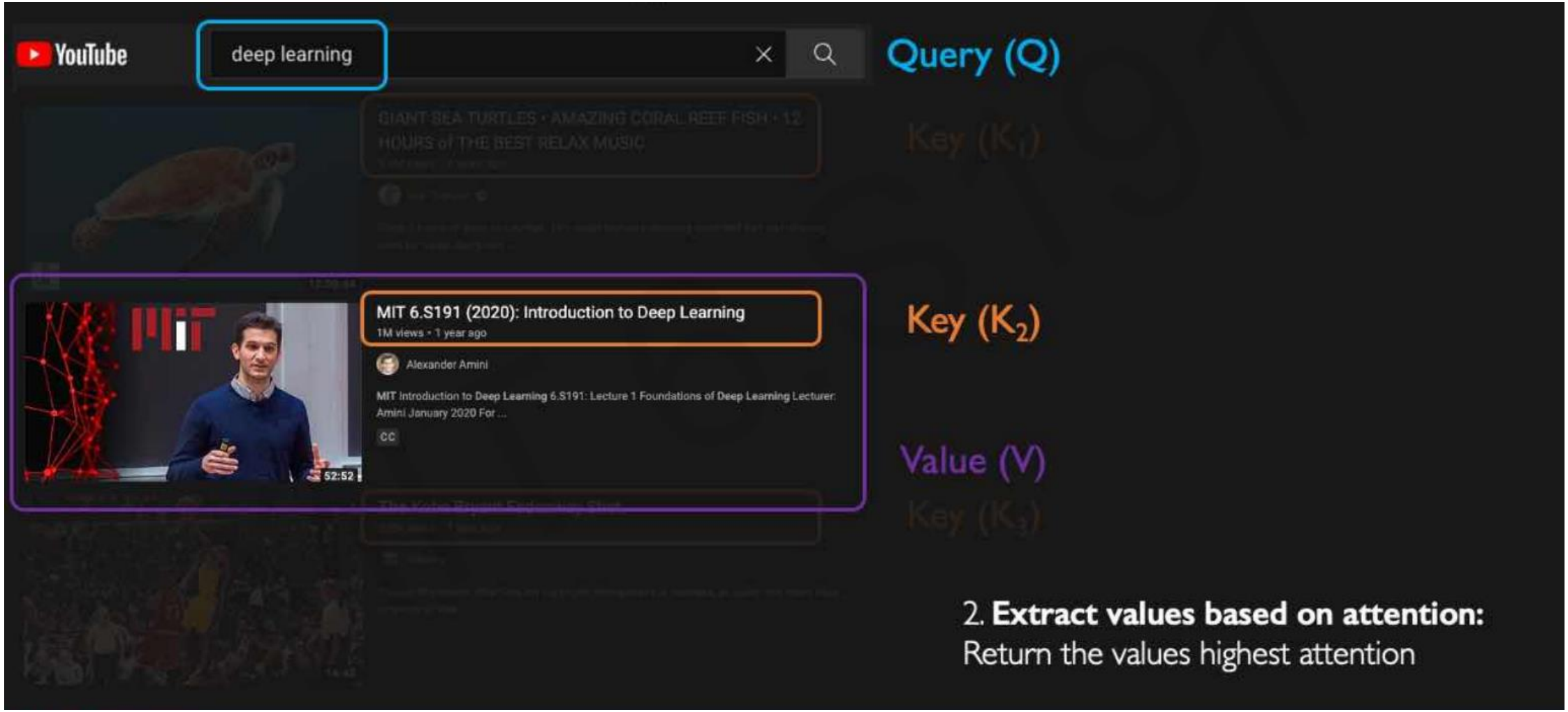
1. Identify which parts to attend to
2. Extract the features with high attention

Similar to a search problem!

Understanding Attention with Search

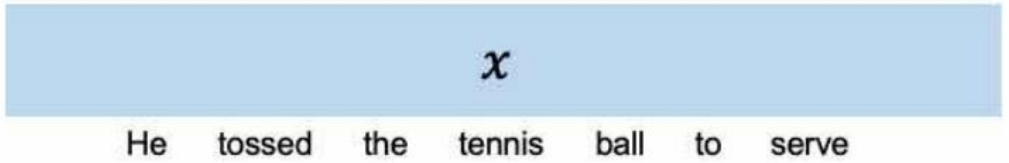


Understanding Attention with Search



Learning Self-Attention with Neural Networks

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



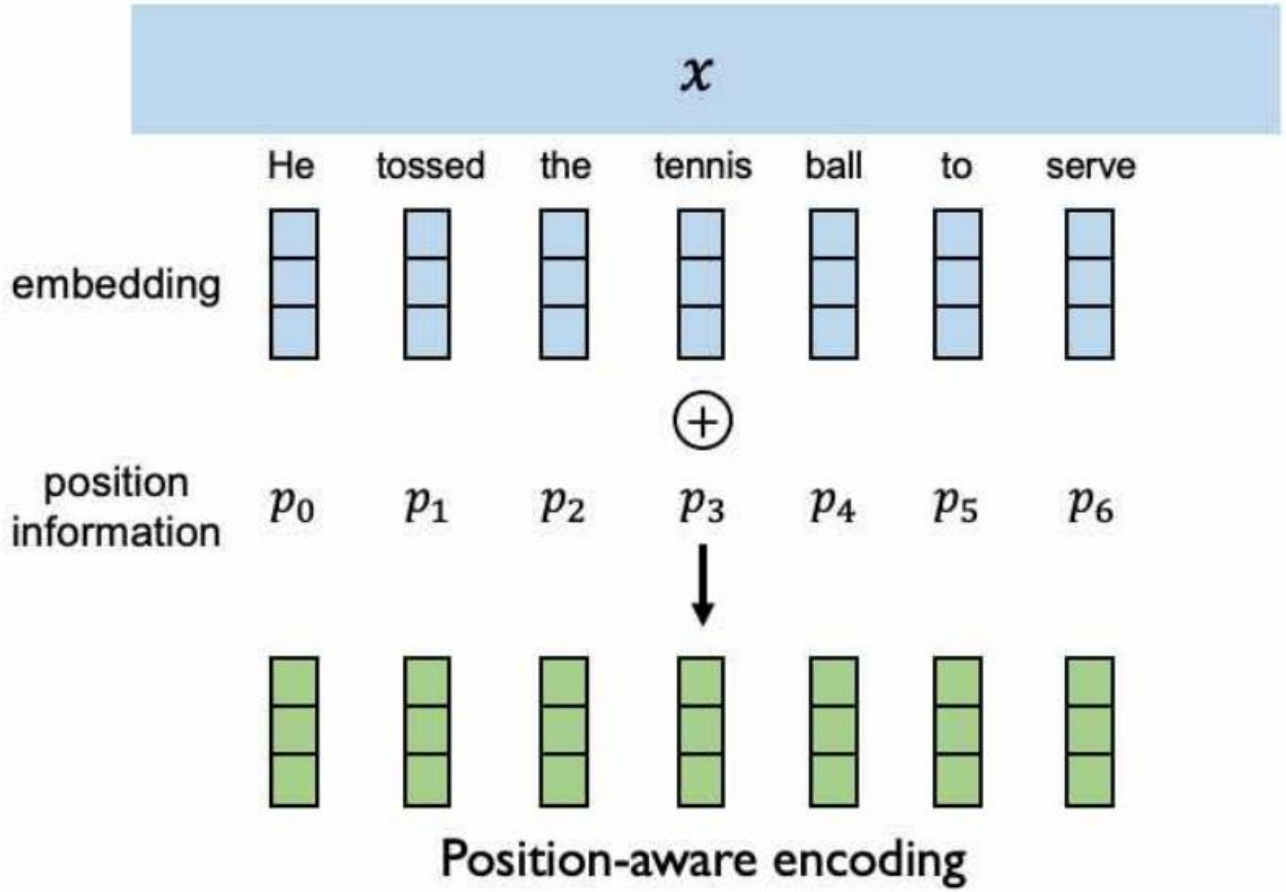
1. Encode **position** information
2. Extract query, key, value for search
3. Compute attention weighting
4. Extract features with high attention

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

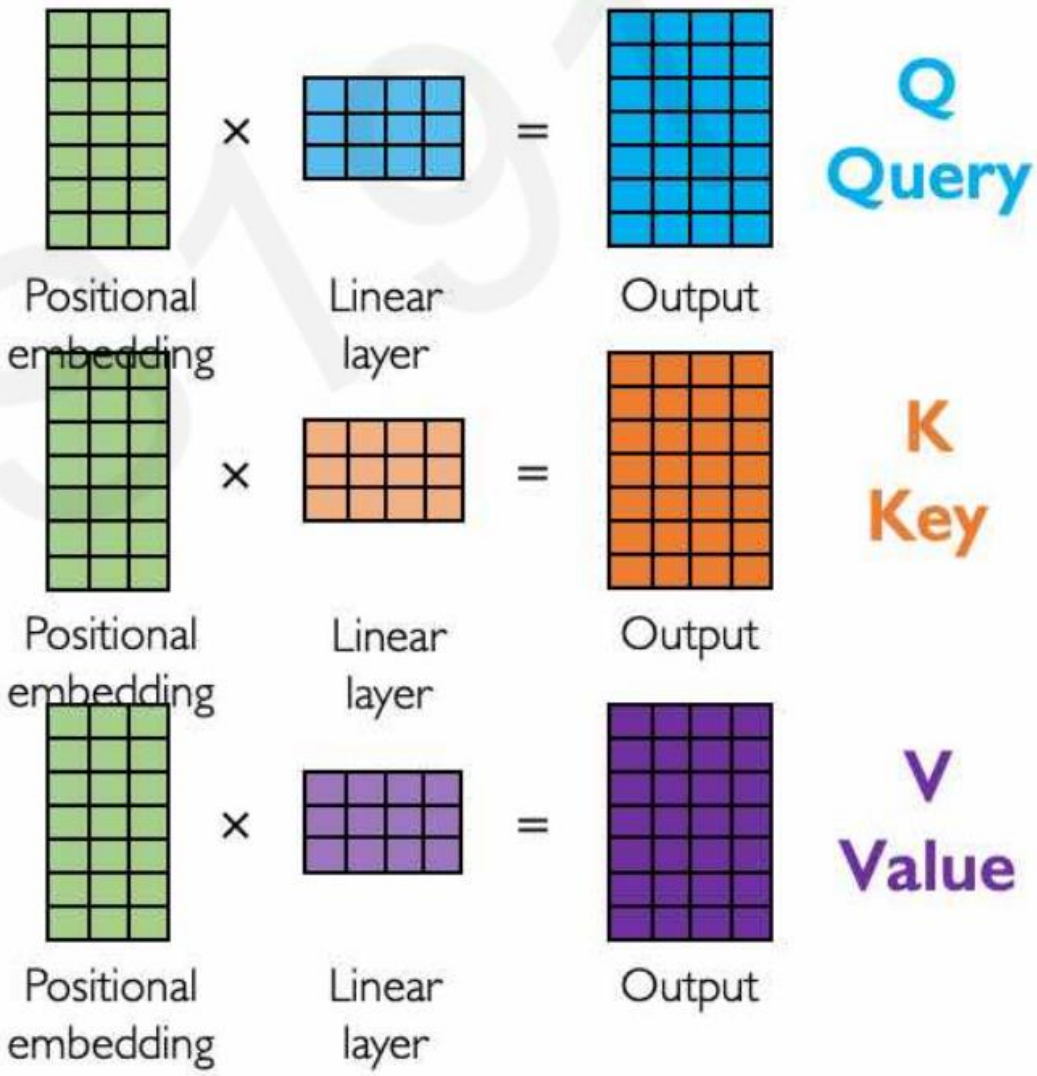


Data is fed in all at once! Need to encode position information to understand order.

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



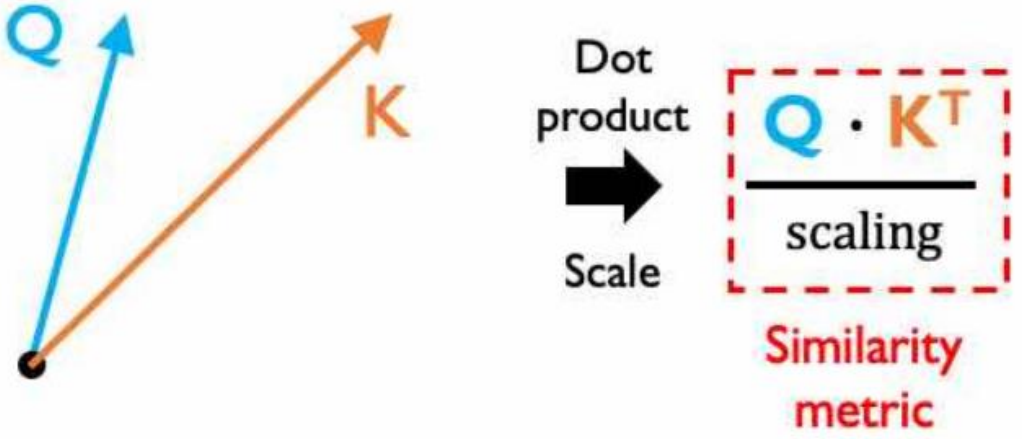
Learning Self-Attention with Neural Networks

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"

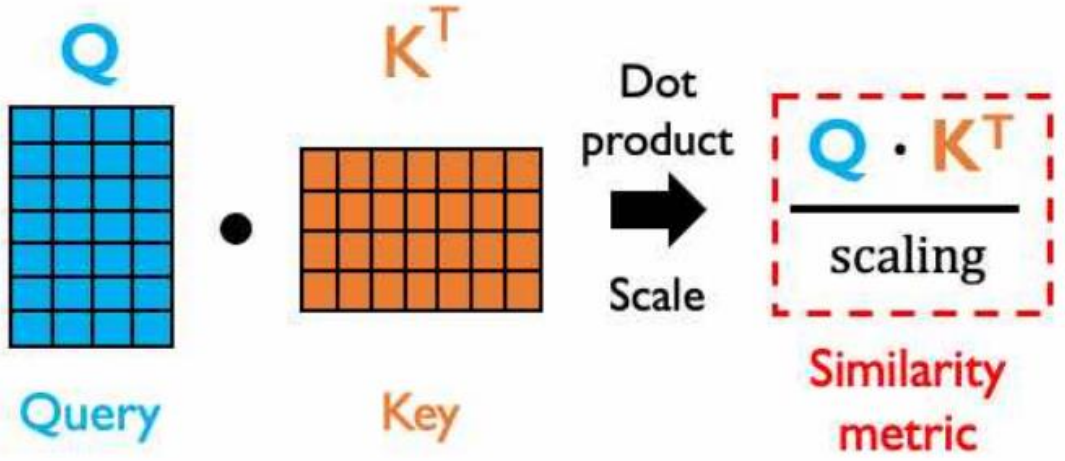
Learning Self-Attention with Neural Networks

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?



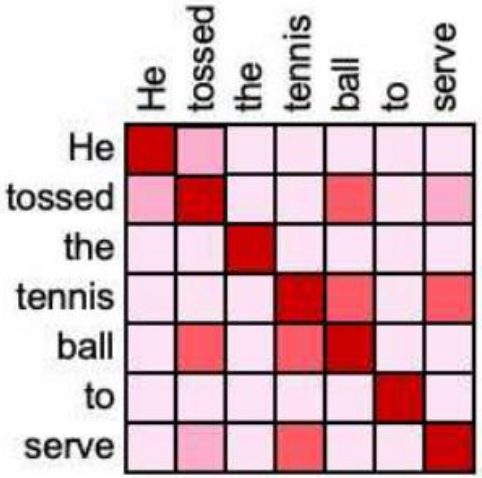
Also known as the "cosine similarity"

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?



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

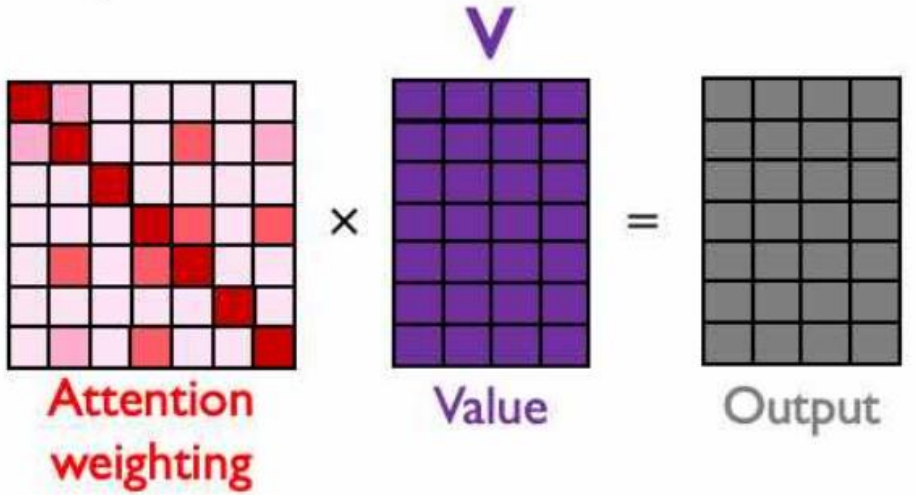
Attention weighting

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



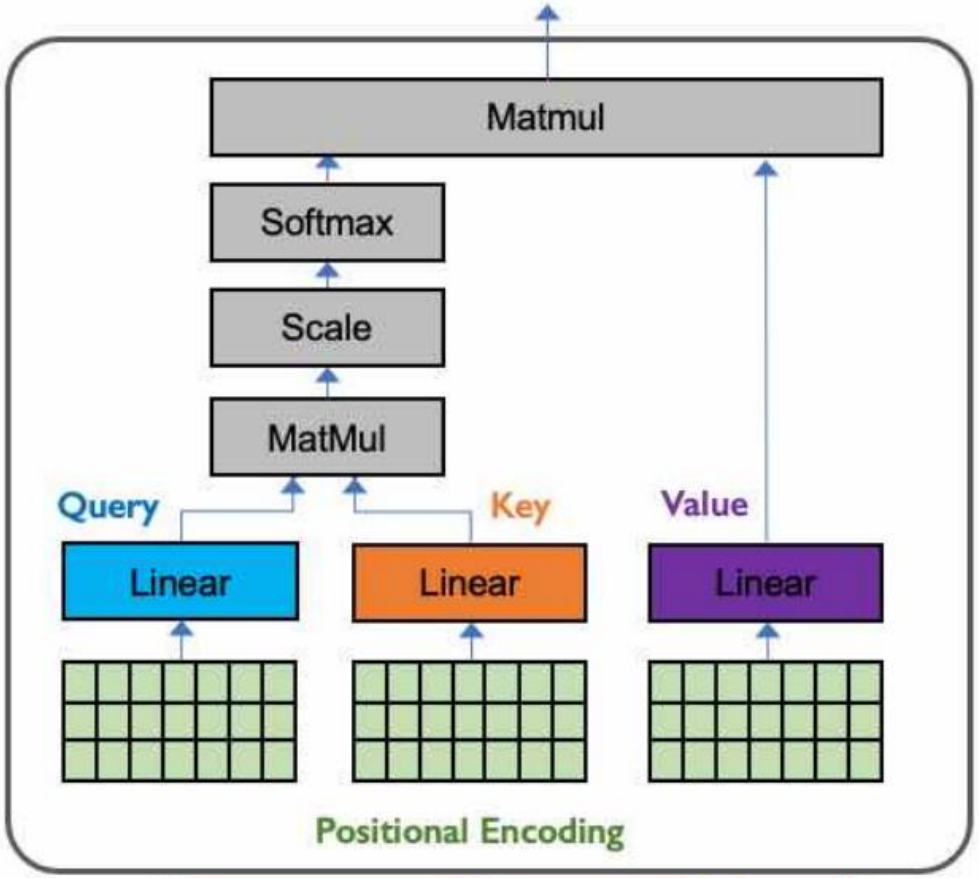
$$softmax\left(\frac{Q \cdot K^T}{scaling}\right) \cdot V = A(Q, K, V)$$

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

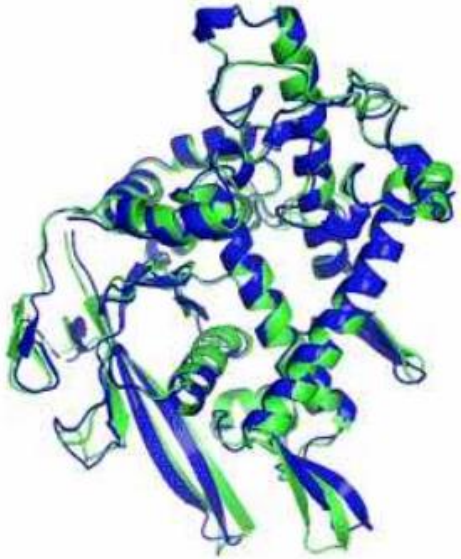
Language Processing

An armchair in the shape of an avocado

BERT, GPT-3

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

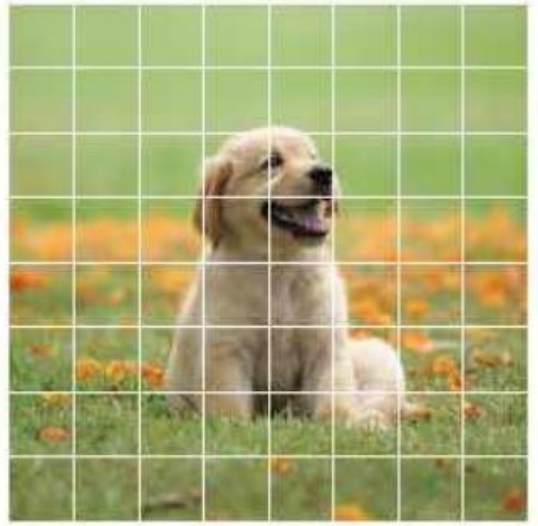
Biological Sequences



AlphaFold2

Jumper et al., *Nature* 2021

Computer Vision



Vision Transformers

Dosovitskiy et al., *ICLR* 2020

KISTI SINCE 1962

2022 TRUST KISTI

Korea Institute of Science and Technology Information