Advanced Recurrent Neural Networks

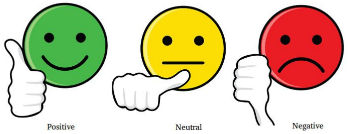
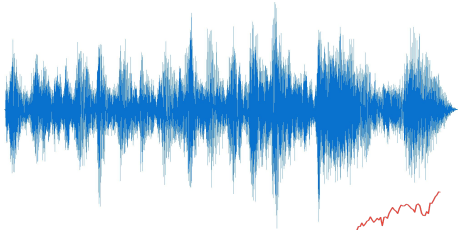
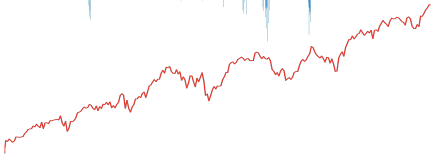
CS344
Deep Learning

WELLESLEY

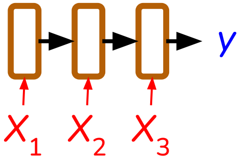
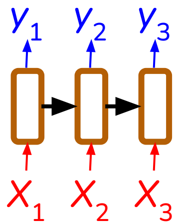
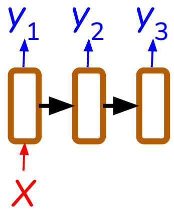
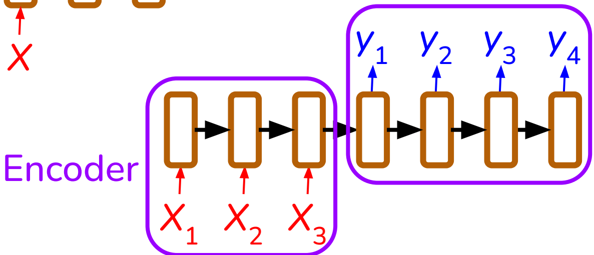
RNNs

- Recap
- GRUs and LSTMs
- Bidirectional
- Attention
- Transformers

Sequence Data

Word labeling	I like red apples	pron verb adj noun
Machine translation	Do you have a pet?	¿Tienes una mascota?
Text generation	Write a poem	Roses are red…
Sentiment classification	Good, cheap food!	
Speech recognition		I stay out too late
Time series prediction		54.7

Different RNNs

Input	Output	Example	Architecture
Sequence	Non-sequence	Sentiment classification	
Sequence	Sequence (same-length)	Word labeling	
Non-sequence	Sequence	Text generation	
Sequence	Sequence (different-length)	Translation	

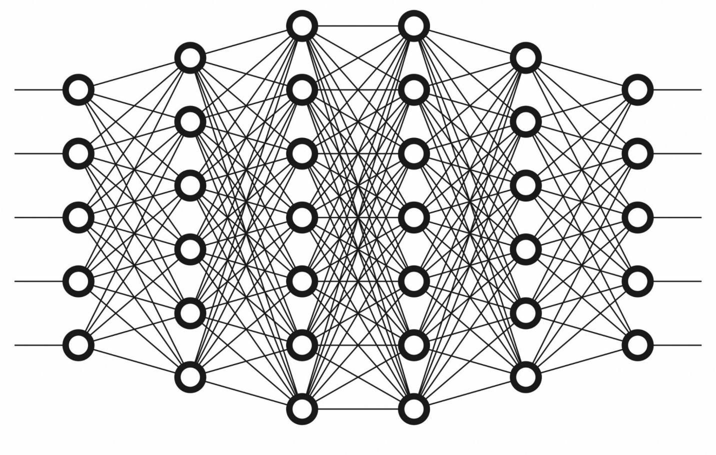
Word Embedding

Apple	College	Ruby	Studying	Fox	Pi
0.52	-1.23	0.16	0.29	0.44	0.4
-0.83	1.42	0.91	0.35	0.06	1.07
0.5	-0.69	-0.55	-0.87	0.16	0.44
1.29	-1.16	1.39	-0.73	0.93	0.64
0.12	0.0	-0.14	-0.08	0.19	0.33
⋮	⋮	⋮	⋮	⋮	⋮
0.27	0.32	-0.25	-0.11	1.51	0.15

Why use *recurrent* NN rather than MLP?

An RNN (like CNN) uses what it's learned about one part of input on other parts of input

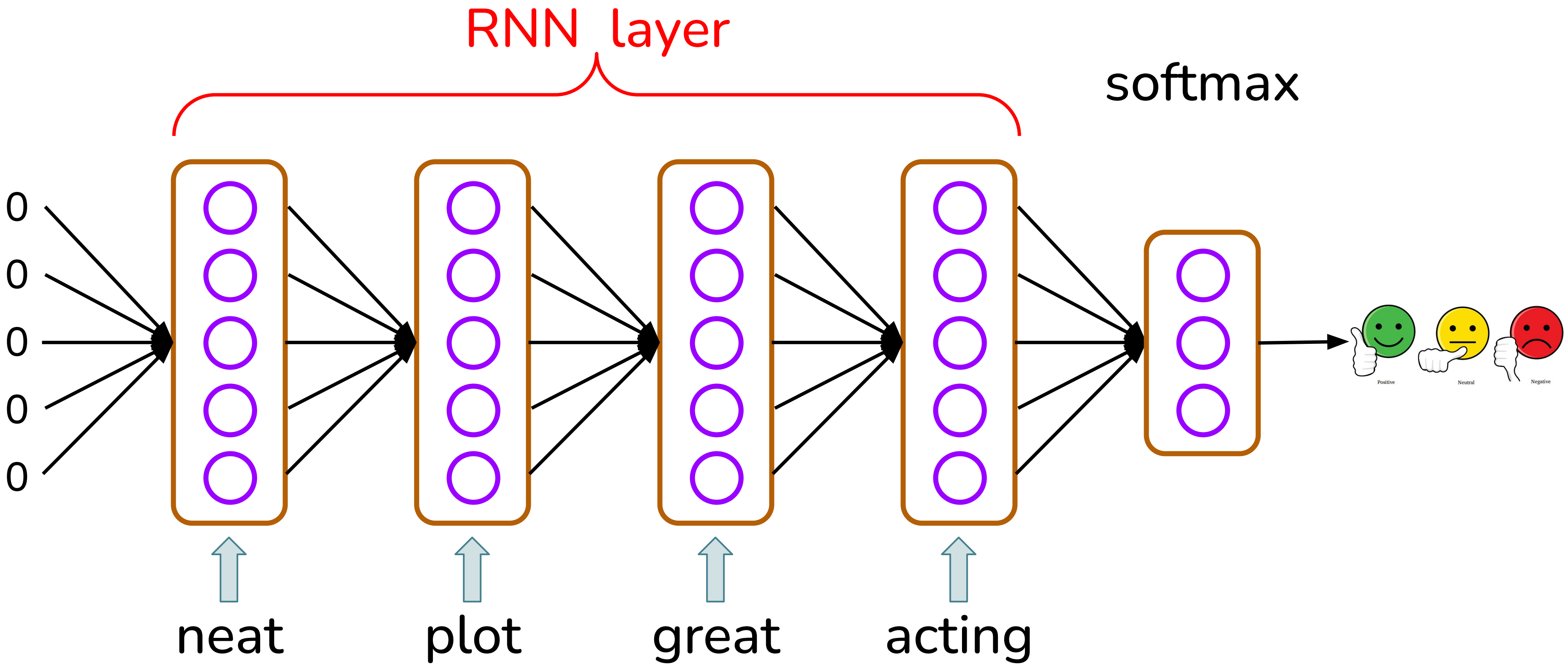
An RNN (like CNN) uses fewer parameters per layer



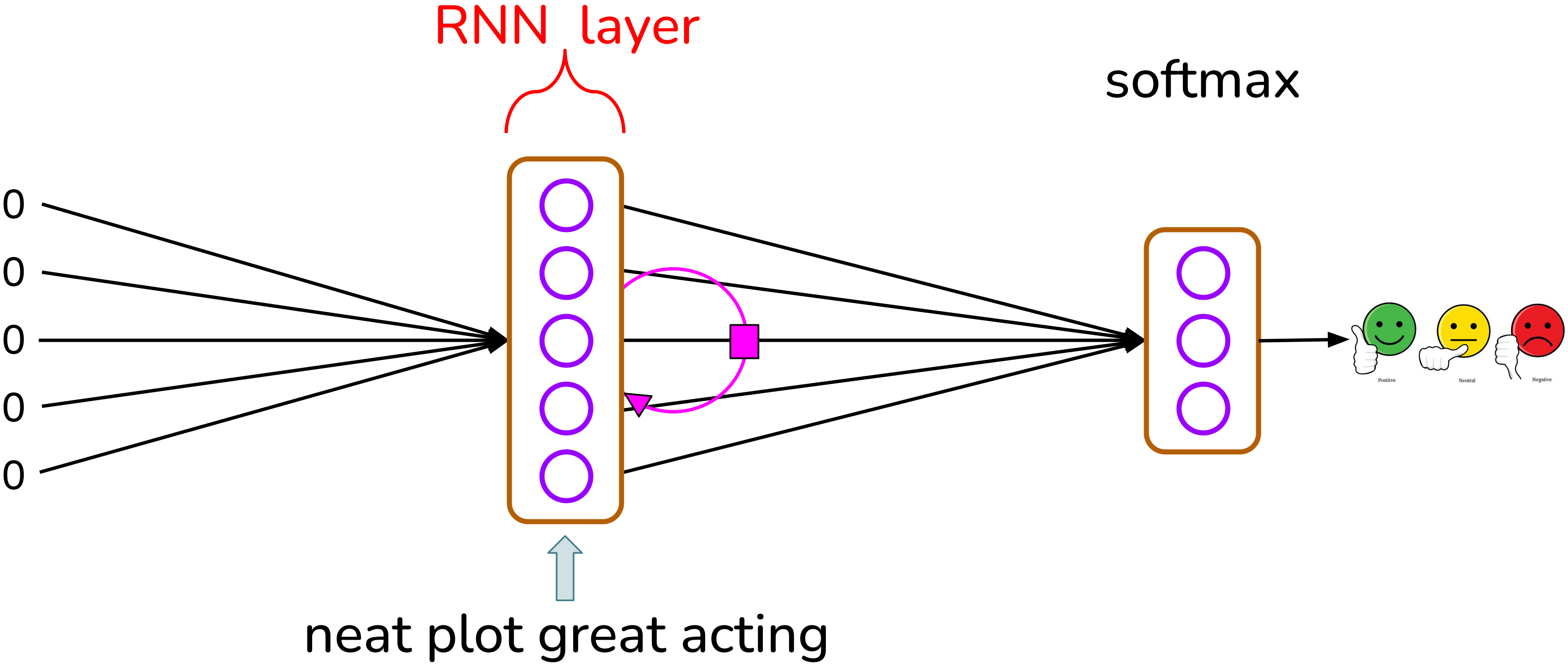
RNN allows for different length inputs and outputs

An RNN is well suited to modeling the sequential nature of data

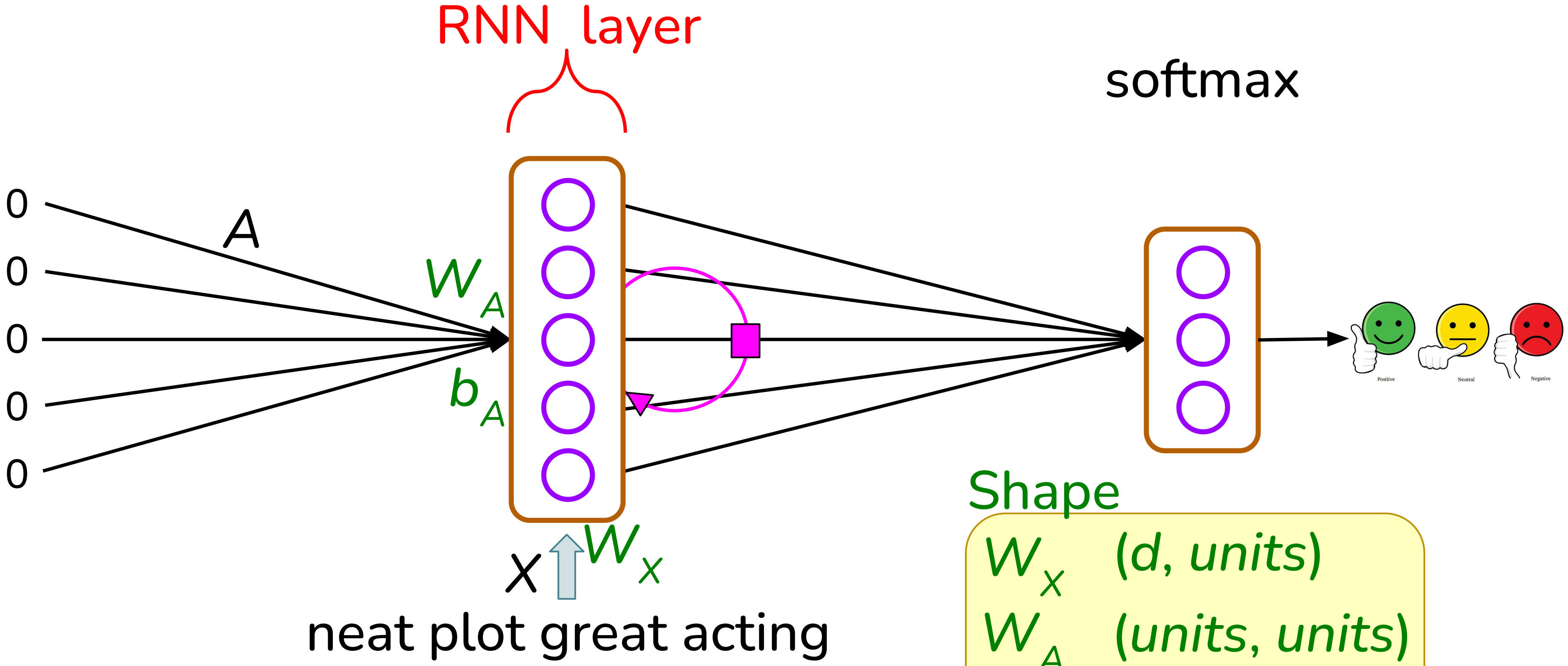
RNN Layer



RNN Layer



RNN Layer *Parameters*



Shape

W_X	$(d, units)$
W_A	$(units, units)$
b_A	$(units,)$

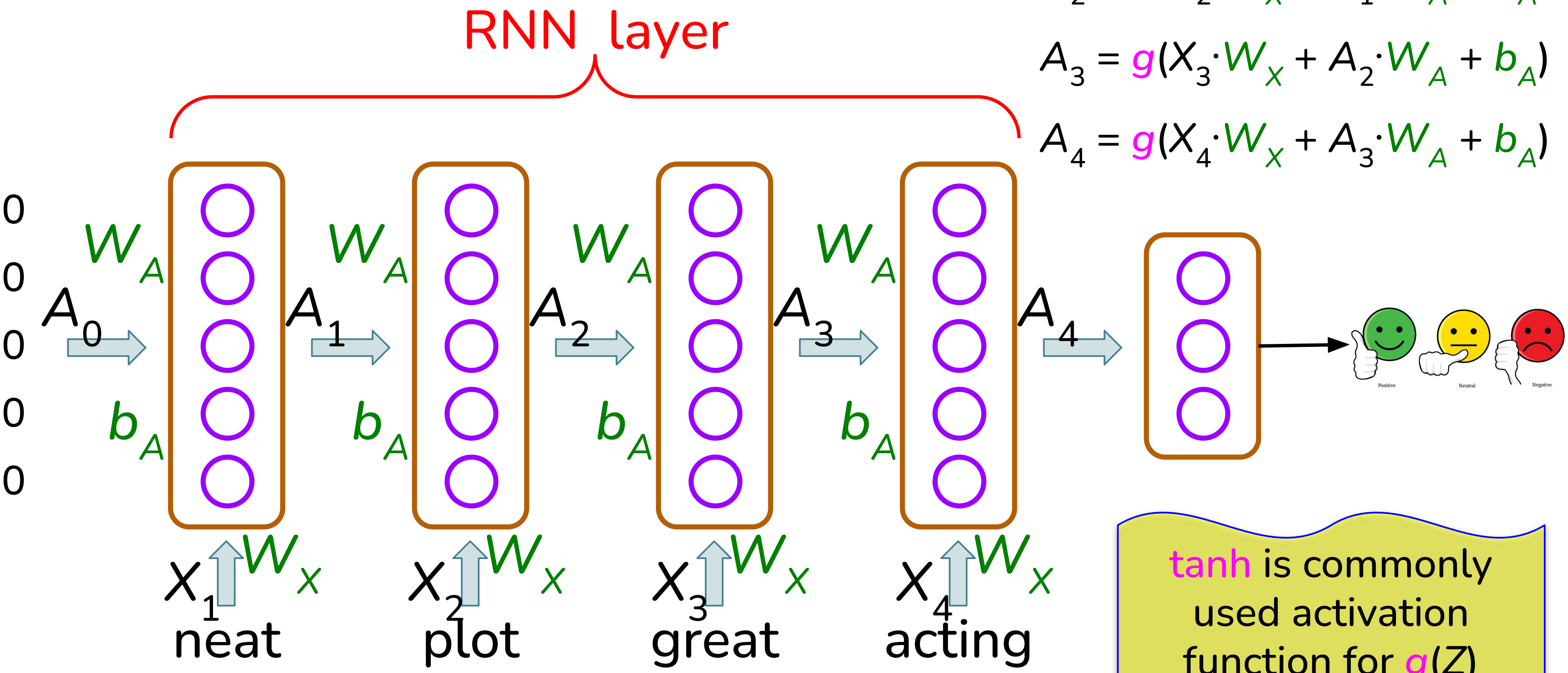
RNN Layer *Parameters*

$$A_1 = g(X_1 \cdot W_X + A_0 \cdot W_A + b_A)$$

$$A_2 = g(X_2 \cdot W_X + A_1 \cdot W_A + b_A)$$

$$A_3 = g(X_3 \cdot W_X + A_2 \cdot W_A + b_A)$$

$$A_4 = g(X_4 \cdot W_X + A_3 \cdot W_A + b_A)$$



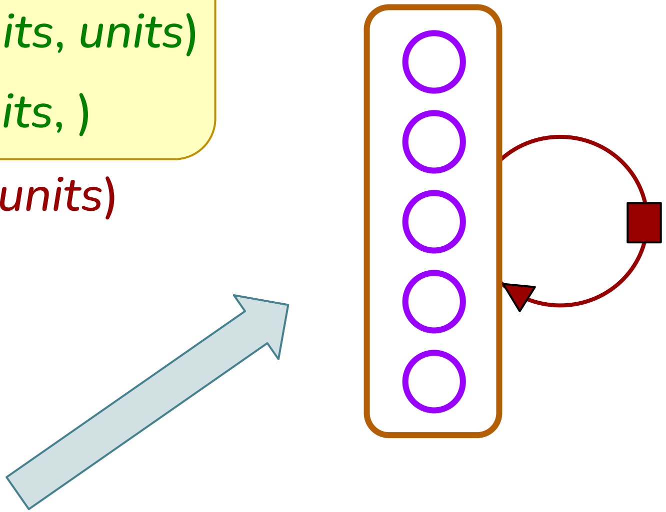
tanh is commonly used activation function for $g(Z)$

RNN Forward Propagation

Shape

W_X	$(d, units)$
W_A	$(units, units)$
b_A	$(units,)$
A	$(1, units)$

RNN layer



$A = [[0\ 0\ 0\ \ldots\ 0]]$

For $t = 0$ to T-1:

$$A = g(X_t \cdot W_X + A \cdot W_A + b_A)$$

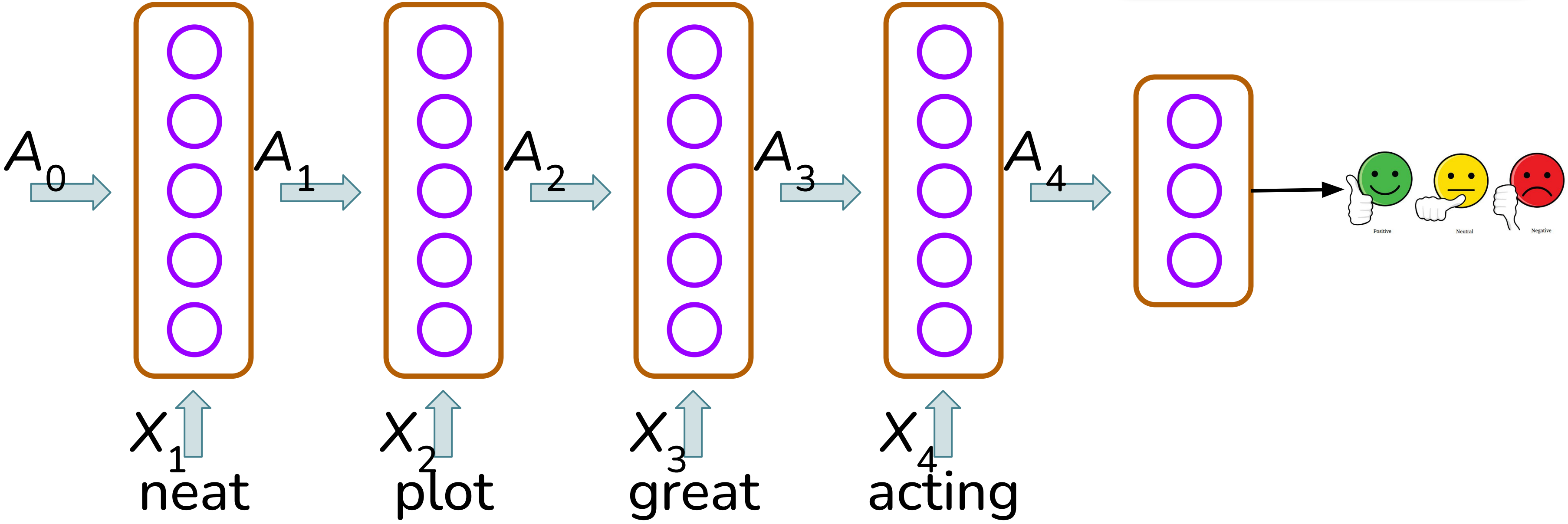
Neat plot. Great acting.

T is number of elements in sequence

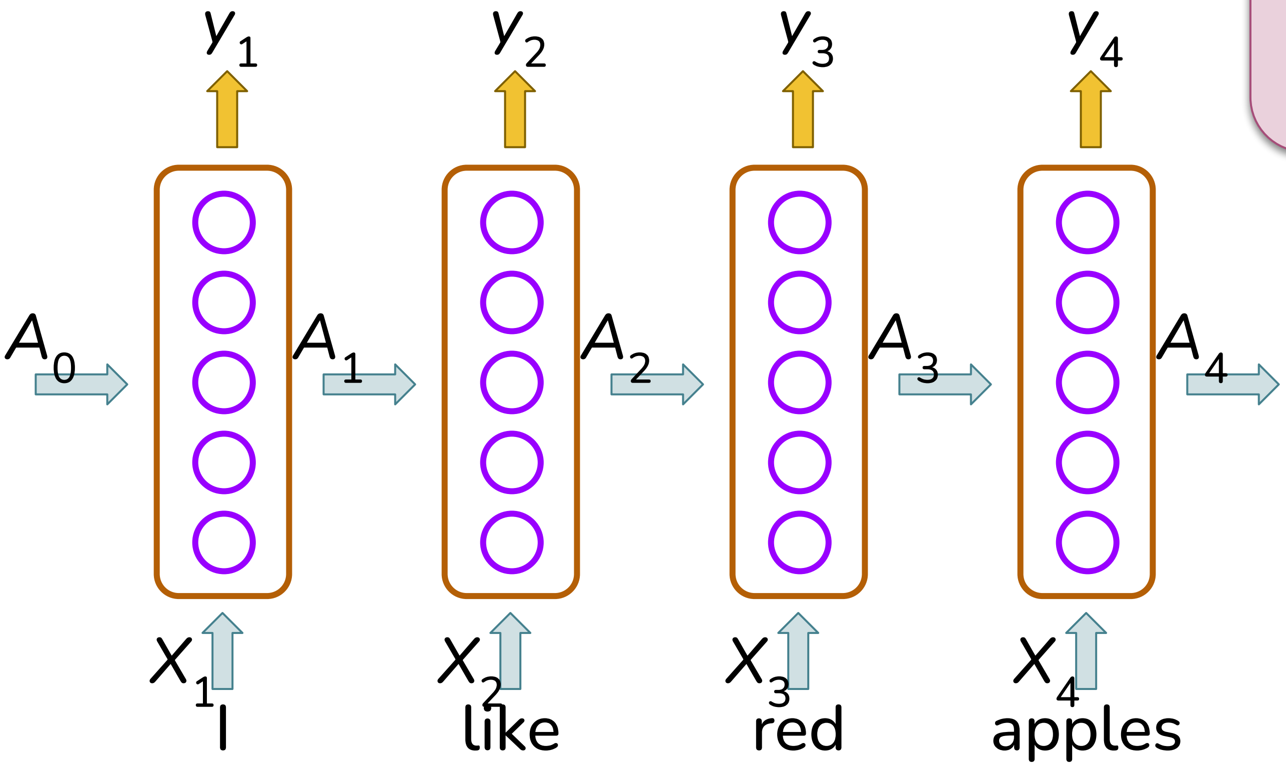
RNN Output

Sentiment Classification

Output is one value, not a sequence



RNN Output



Word Labeling

Output is sequence, one value per input

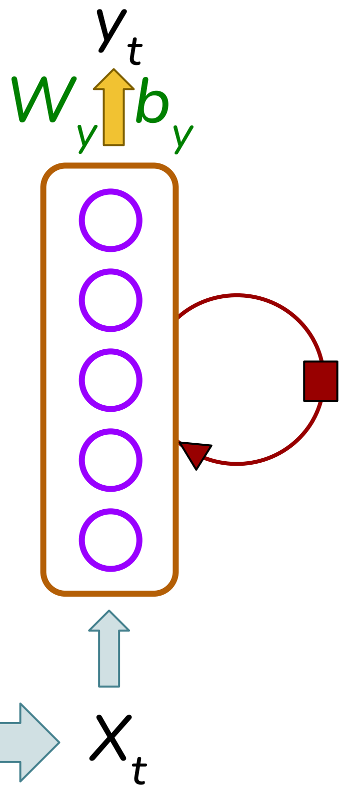
$$y_t = g(A \cdot W_y + b_y)$$

RNN Output *Parameters*

Shape

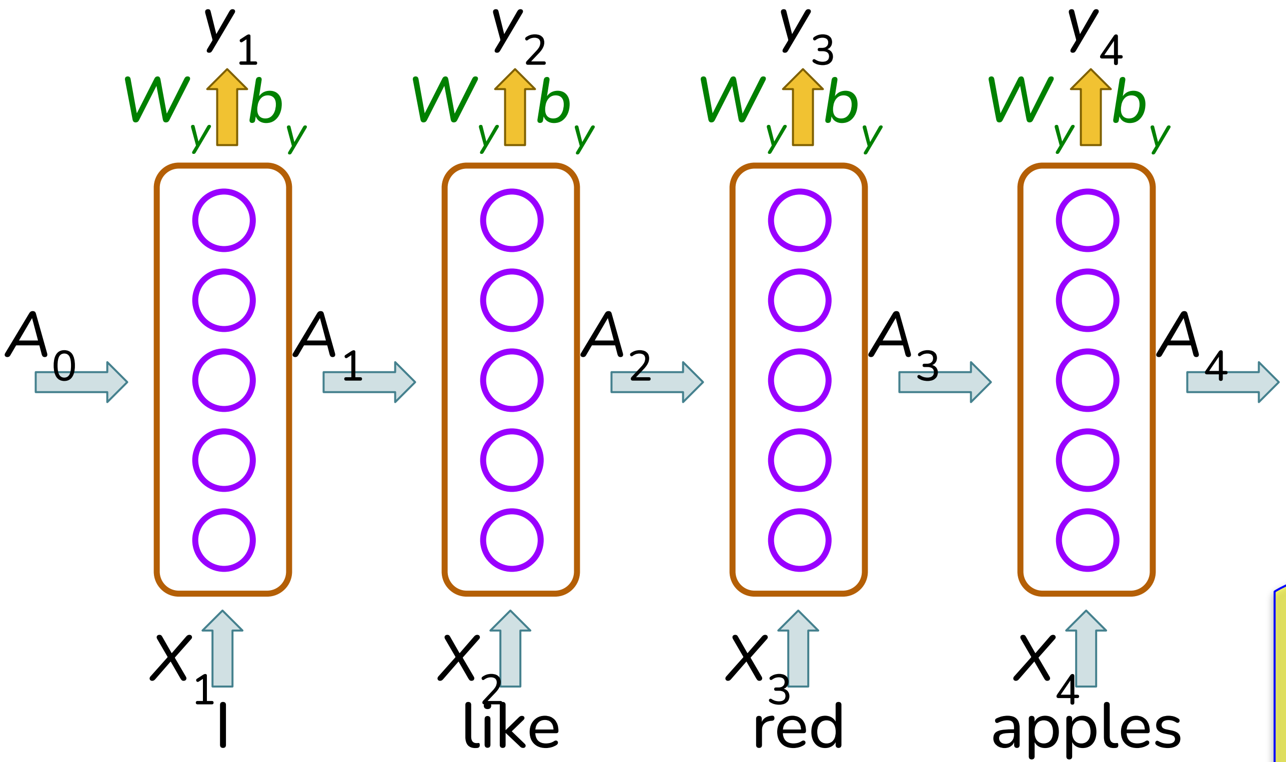
W_y	$(units, ?)$
b_y	$(?,)$

I like red apples



Activation function g depends on problem

RNN Output



$$y_1 = g(A_1 \cdot W_y + b_y)$$

$$y_2 = g(A_2 \cdot W_y + b_y)$$

$$y_3 = g(A_3 \cdot W_y + b_y)$$

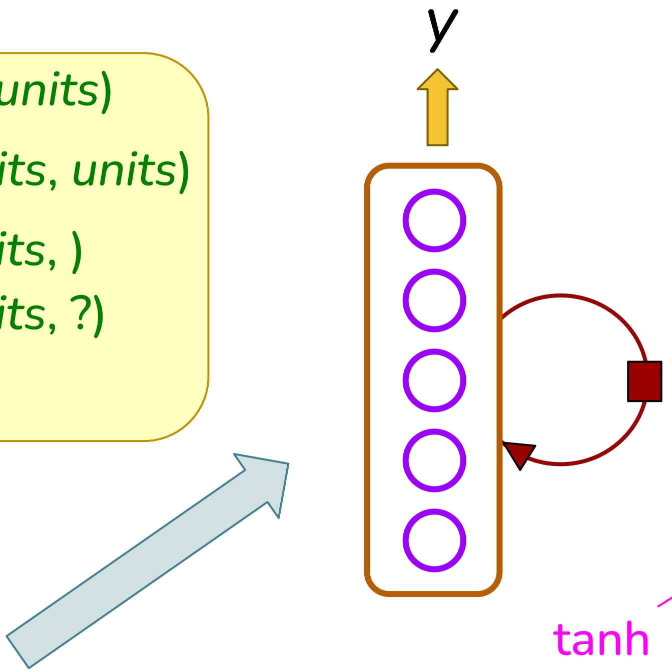
$$y_4 = g(A_4 \cdot W_y + b_y)$$

Activation function g depends on problem

RNN Forward Propagation

Shape

W_X	(*d*, *units*)
W_A	(*units*, *units*)
b_A	(*units*,)
W_y	(*units*, ?)
b_y	(?,)



I like red apples

A = [[0 0 0 … 0]] **# units**

y = [0 0 … 0] **# T**

For t = 0 to T-1:

$$A = g(X_t \cdot W_X + A \cdot W_A + b_A)$$

$$y_t = g(A \cdot W_y + b_y)$$

tanh

sigmoid, softmax, linear

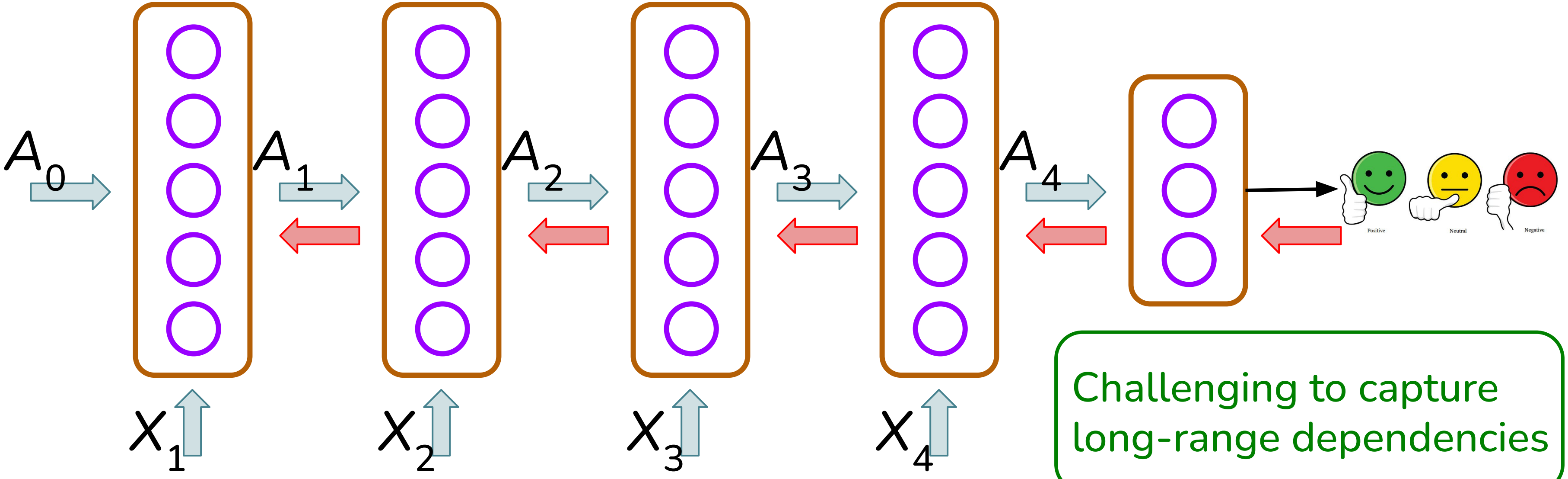
T is number of elements in sequence

RNNs

- Recap
- GRUs and LSTMs
- Bidirectional
- Attention
- Transformers

Vanishing Gradients

For deeper networks, it can be difficult for the gradient (error) at end to propagate back to affect earlier layers

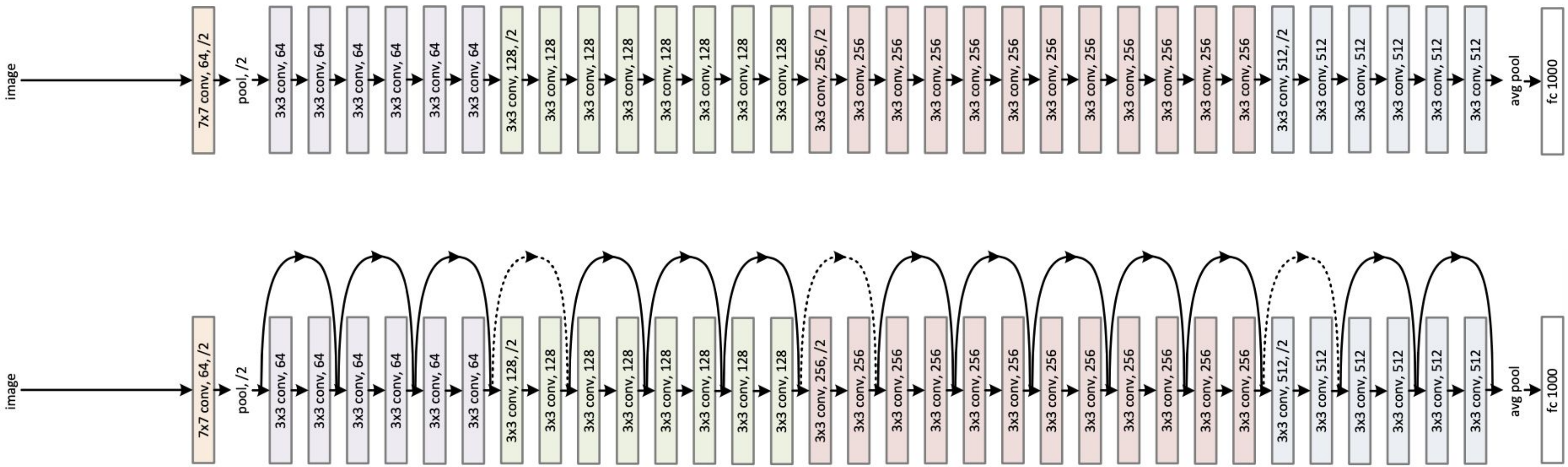


Challenging to capture long-range dependencies

student takes is

The students who take CS344 at Wellesley College are having a deep learning experience.

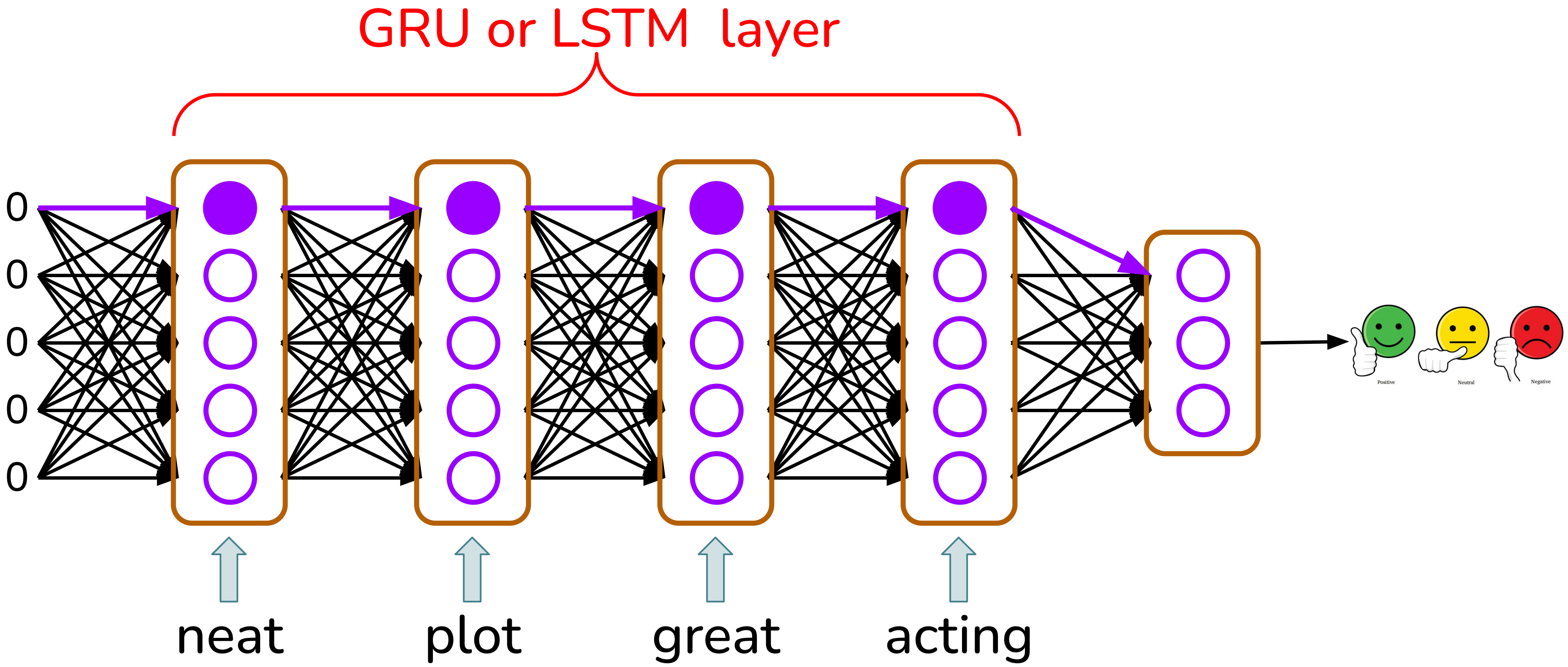
ResNet (Residual Network)



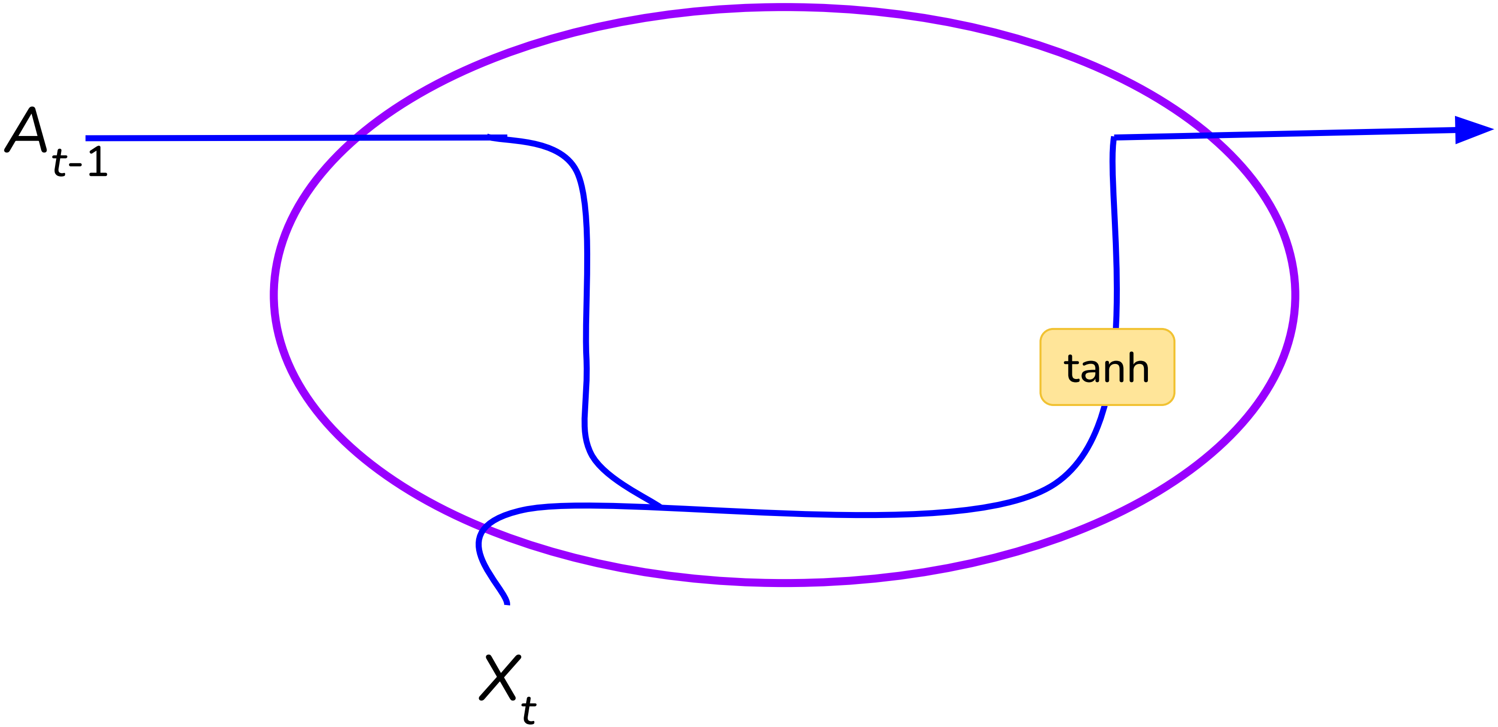
GRUs and LSTMs

- A *GRU* (Gated Recurrent Unit) or *LSTM* (Long short-term memory) is similar to the simple RNN unit
- GRUs and LSTMs have extra sets of parameters, called *gates*: GRU (2 gates), LSTM (3 gates)
- Gates enable units to simulate *memory*, i.e., a unit can output newly computed values and/or pass along (as output) what it received as input
- GRUs and LSTMs help *address the vanishing gradient problem* and *capture longer range dependencies*

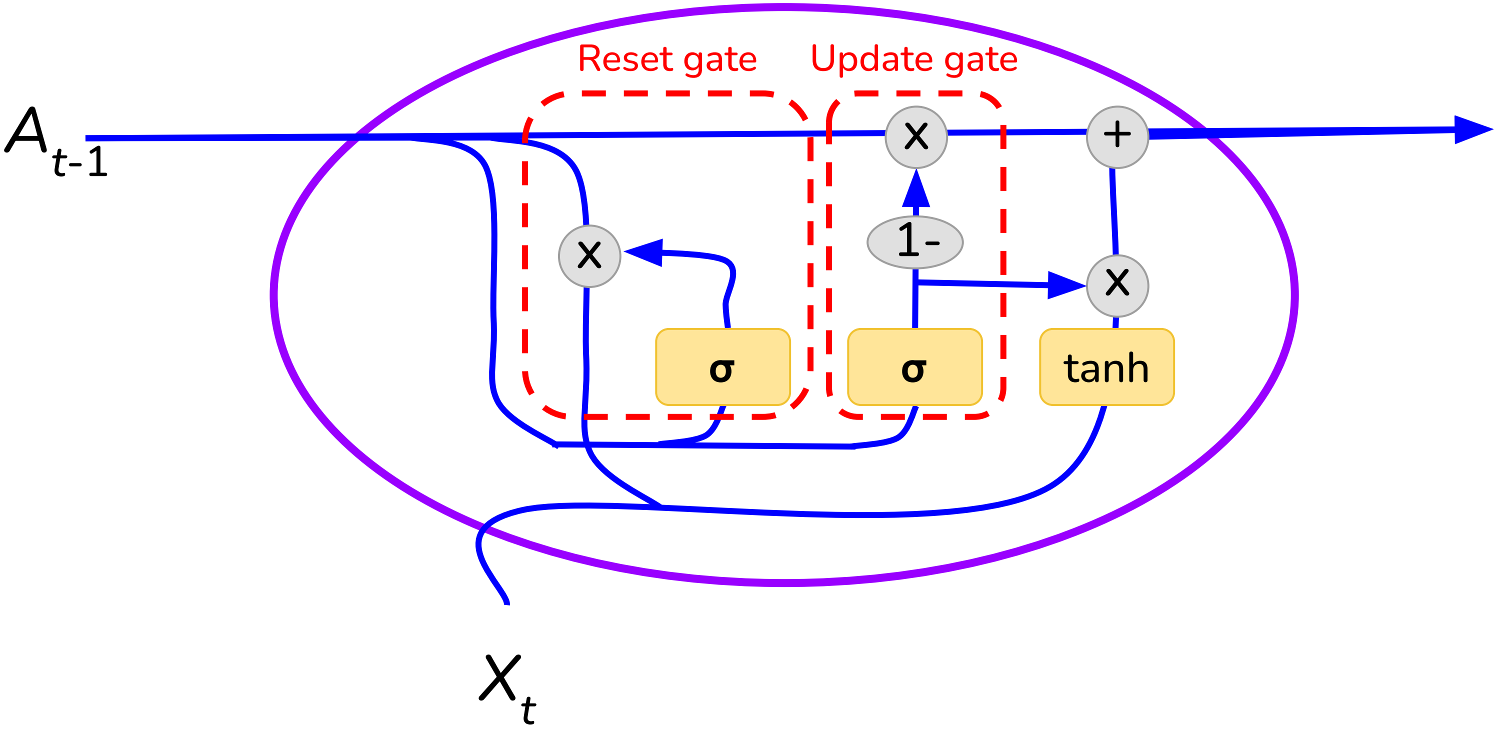
GRUs and LSTMs



Simple RNN Unit



GRU



RNNs

- Recap
- GRUs and LSTMs
- Bidirectional
- Attention
- Transformers

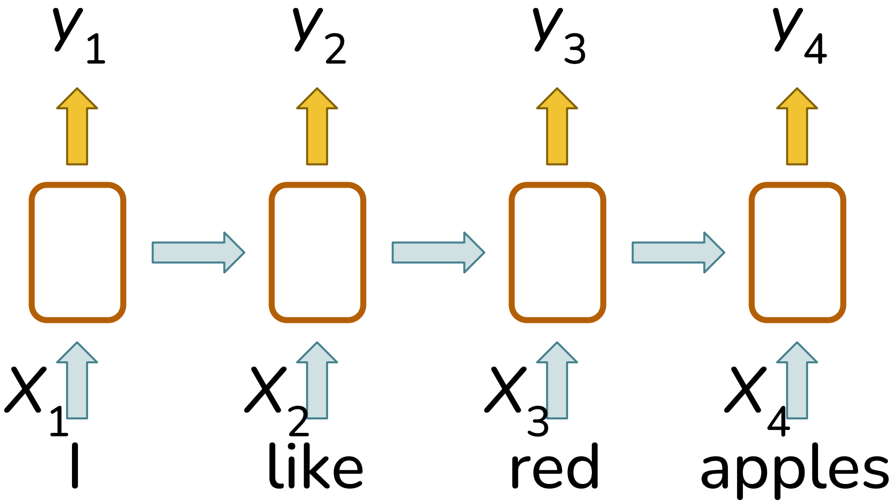
Unidirectional

NOUN
Rest and relaxation are healthy

Rest before your next class
VERB

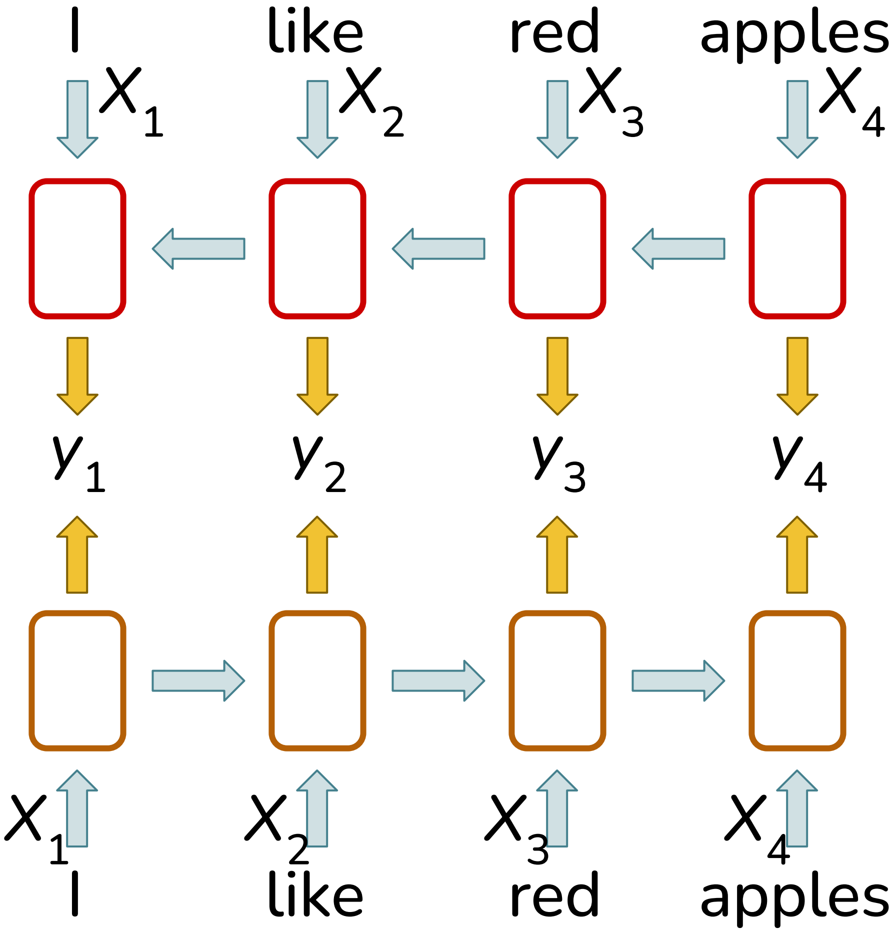
NOUN
Pass me the orange

Pass me the orange marker
ADJ

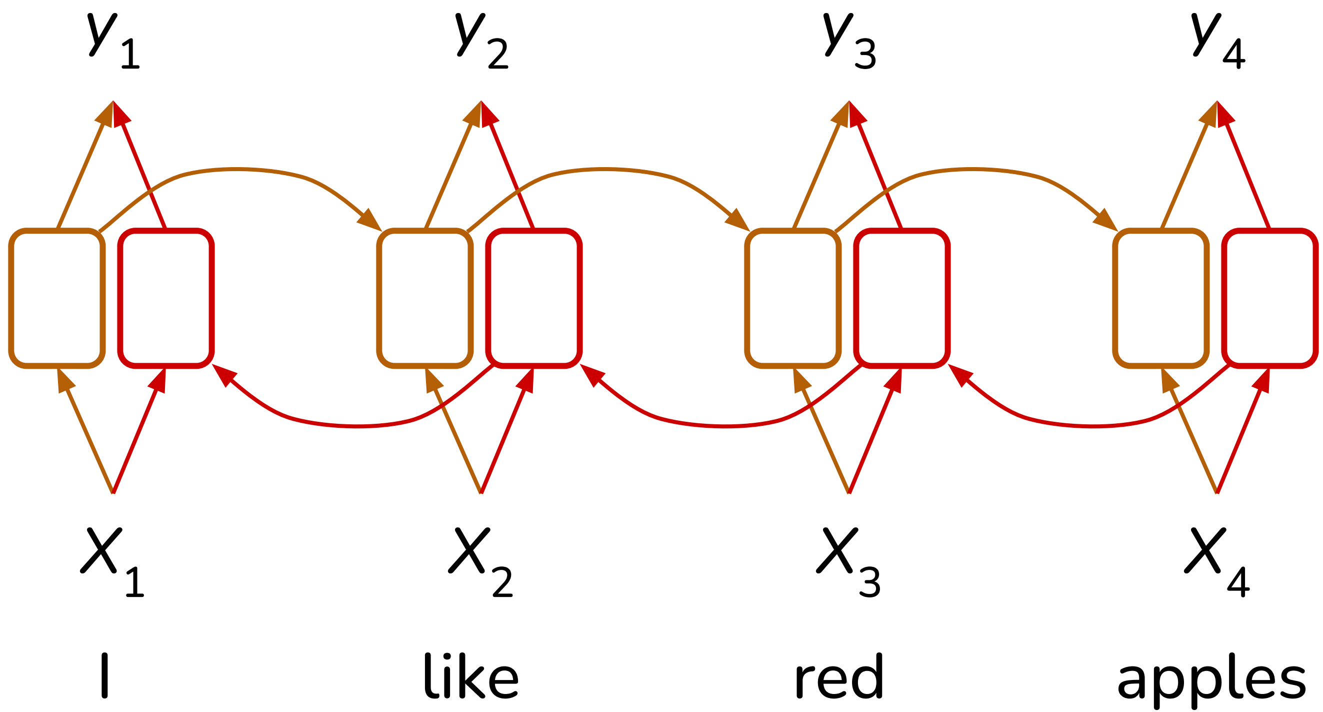


Bidirectional

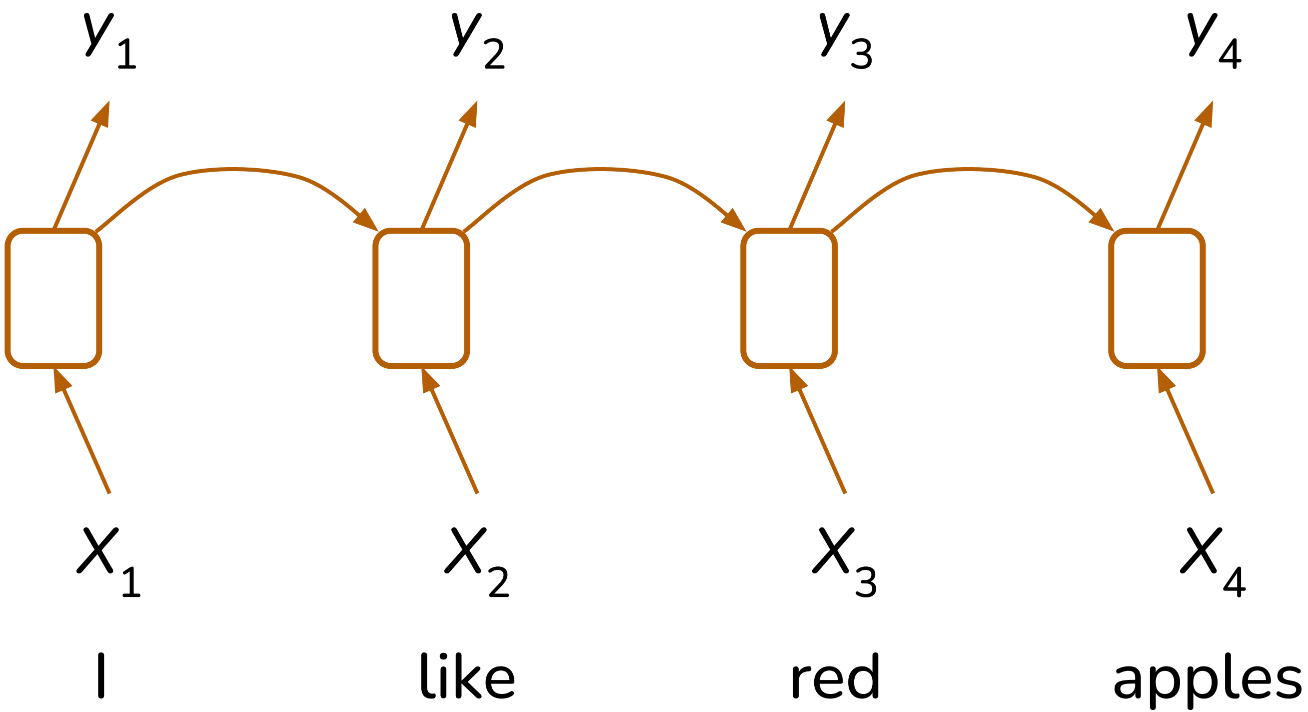
Simple RNN
GRU
LSTM



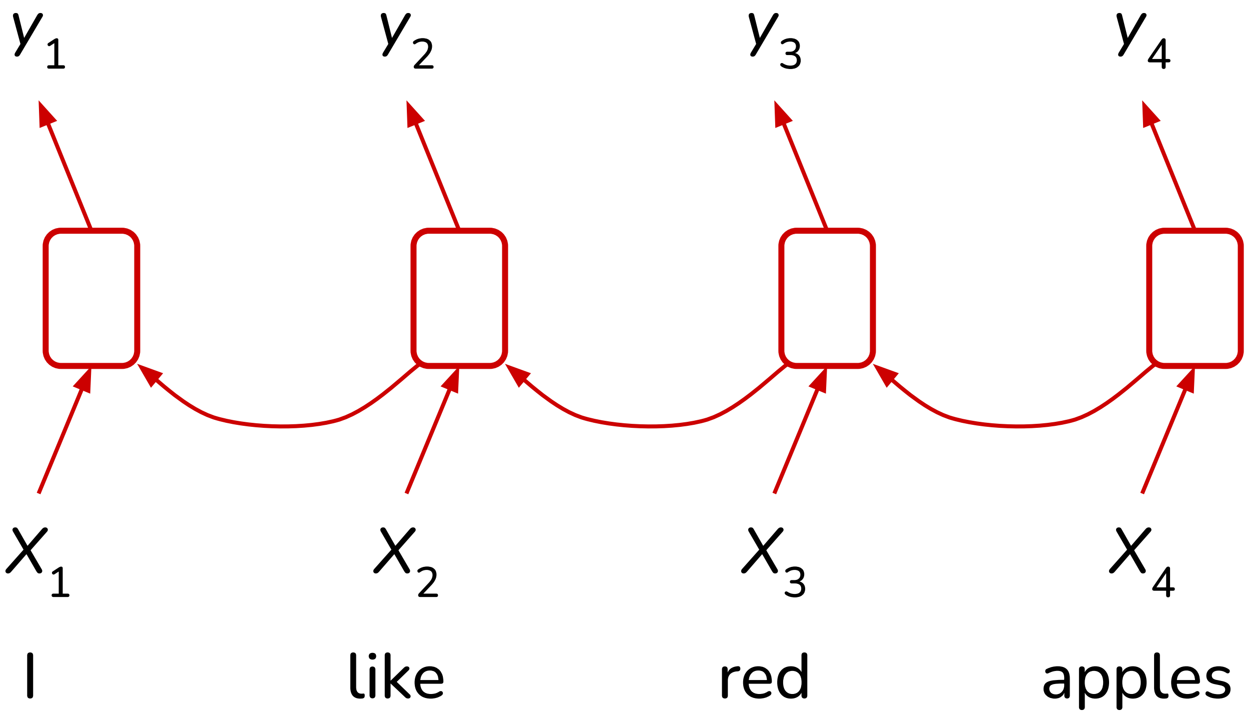
Bidirectional



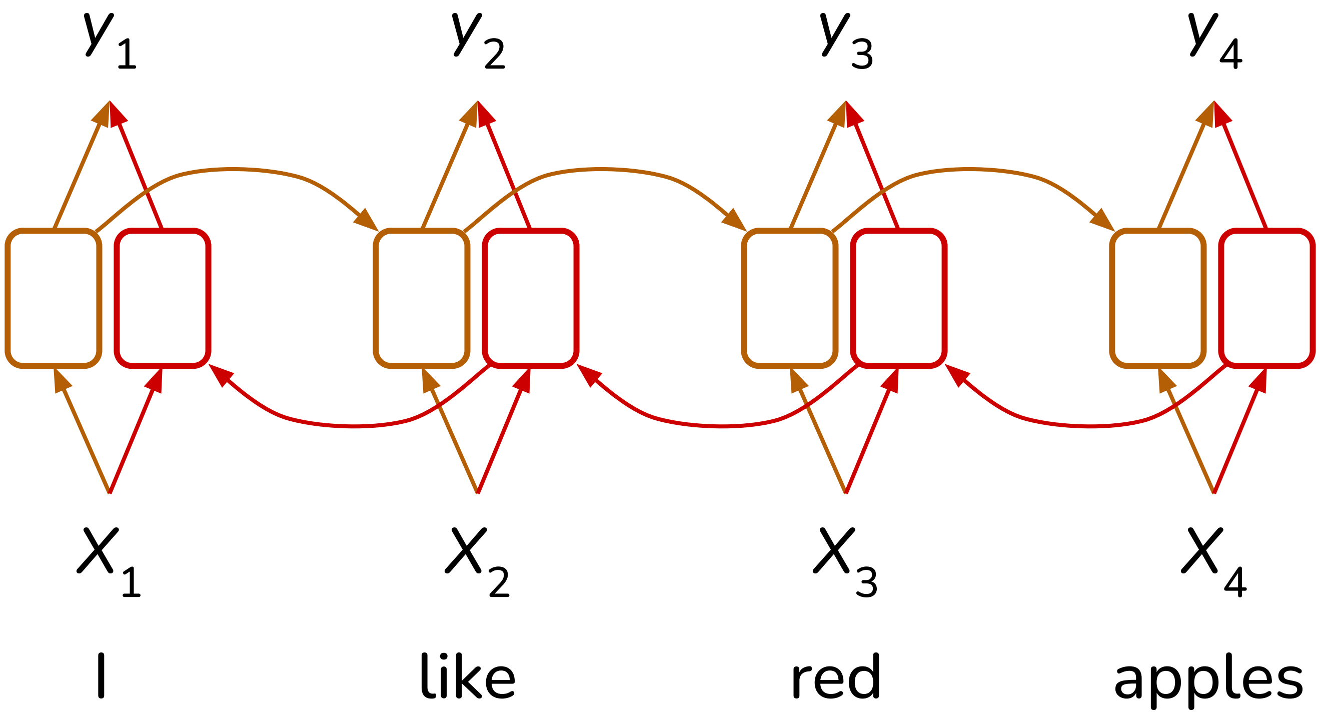
Bidirectional



Bidirectional

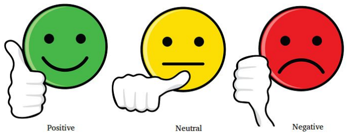


Bidirectional

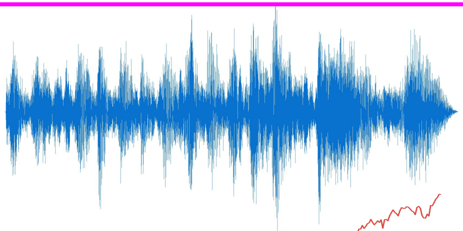
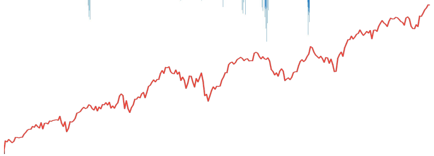


Unidirectional and Bidirectional

Bidirectional

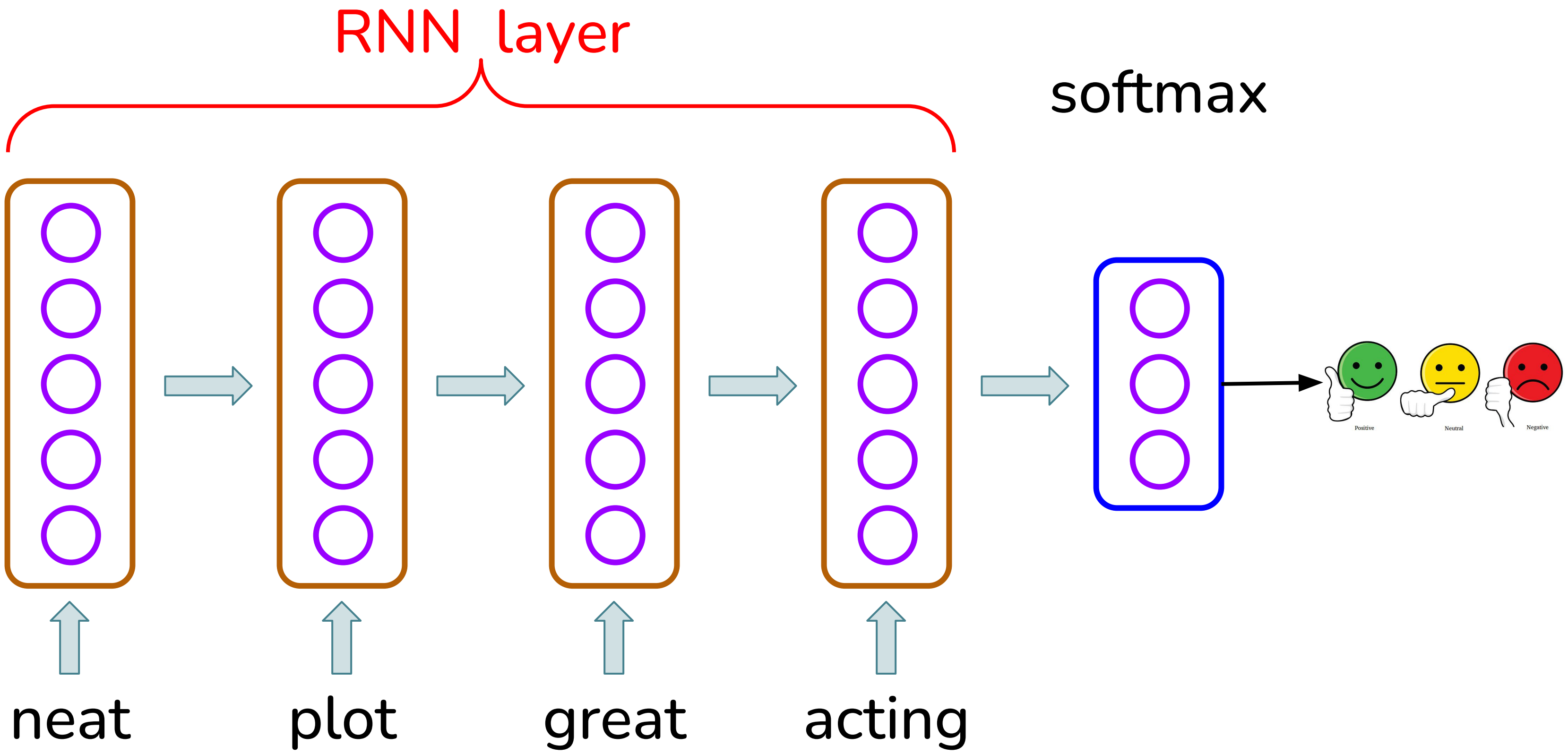
Word labeling	I like red apples	pron verb adj noun
Machine translation	Do you have a pet?	¿Tienes una mascota?
Sentiment classification	Good, cheap food!	

Unidirectional

Speech recognition		I stay out too late
Time series prediction		54.7

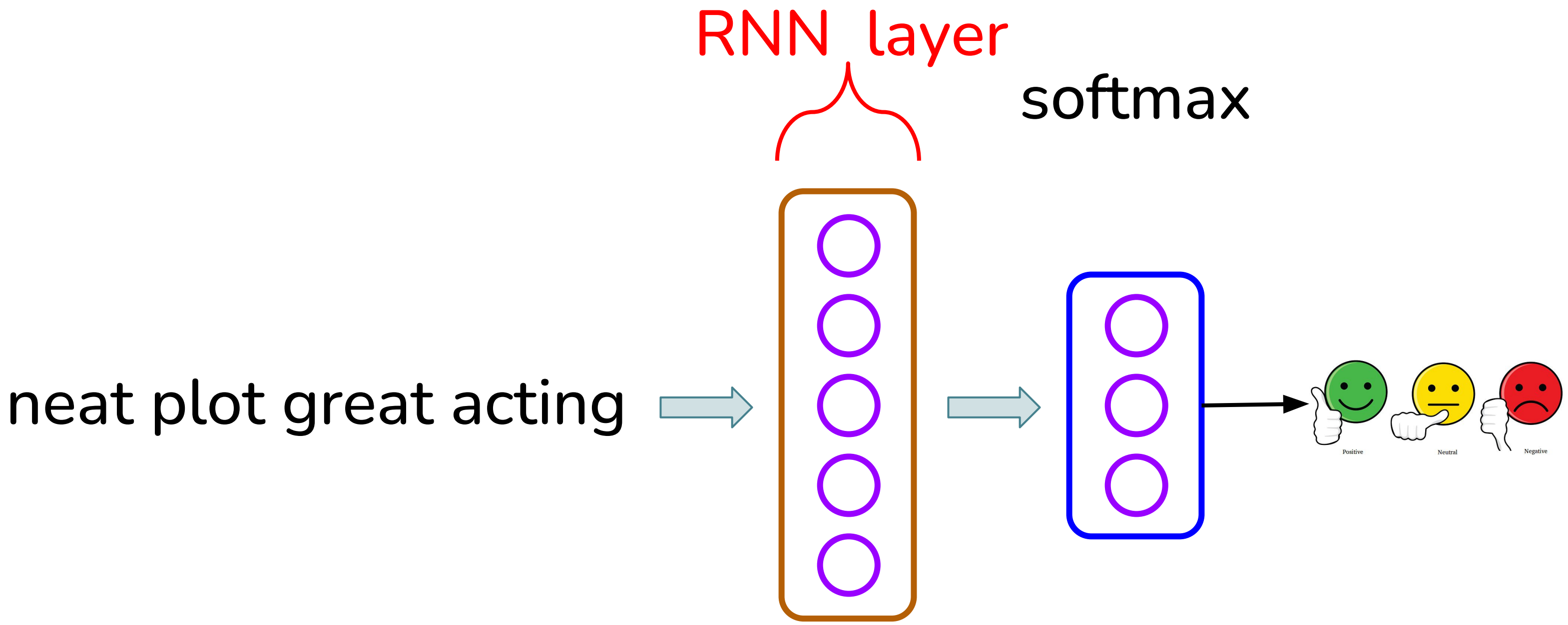
RNNs

```
SimpleRNN(5)
Dense(3, activation='softmax')
```

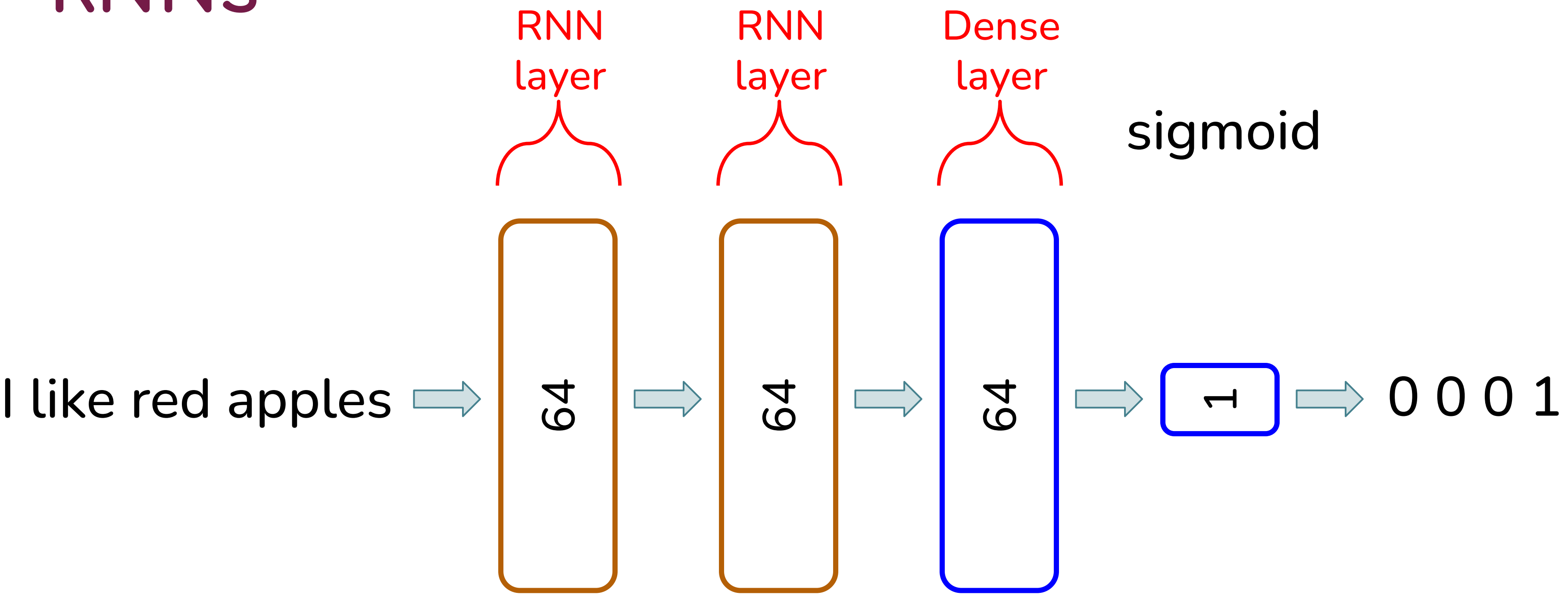


RNNs

```
SimpleRNN(5)
Dense(3, activation='softmax')
```



RNNs

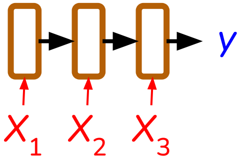
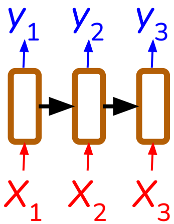
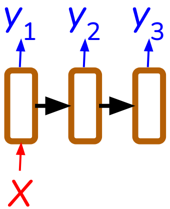
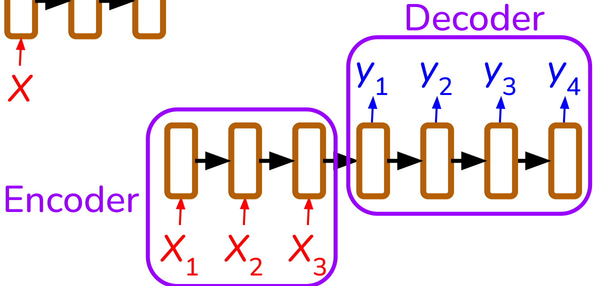



```
Bidirectional(LSTM(64, return_sequences=True))
Bidirectional(LSTM(64, return_sequences=True))
Dense(64, activation='relu')
Dense(1, activation='sigmoid')
```

RNNs

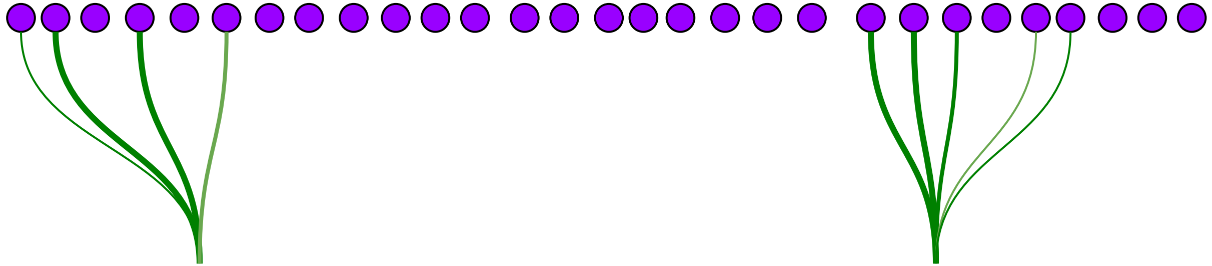
- Recap
- GRUs and LSTMs
- Bidirectional
- Attention
- Transformers

Different RNNs

Input	Output	Example	Architecture
Sequence	Non-sequence	Sentiment classification	
Sequence	Sequence (same-length)	Word labeling	
Non-sequence	Sequence	Text generation	
Sequence	Sequence (different-length)	Translation	

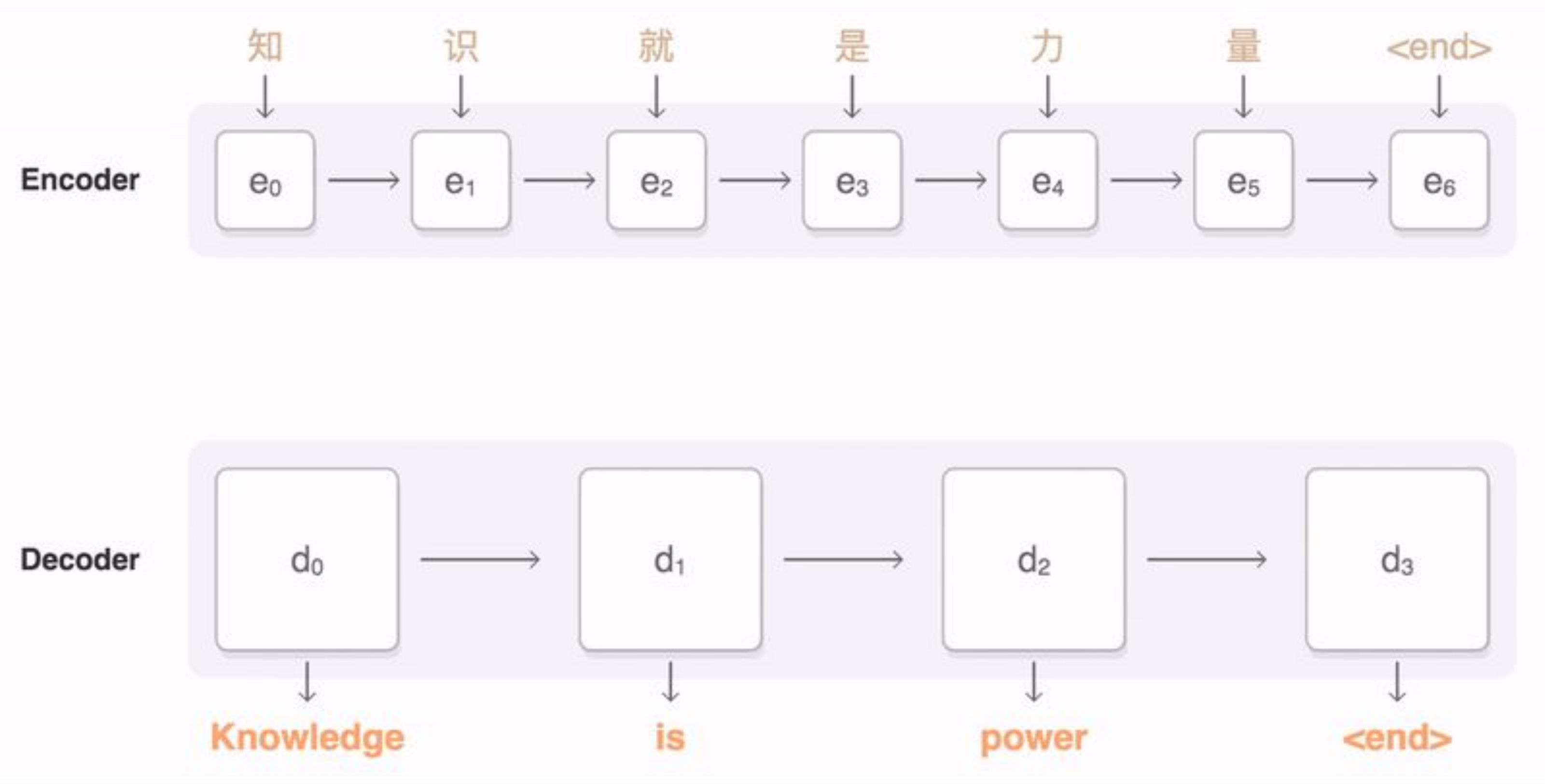
Attention

เธอ**ต้องออกจากบ้าน**เดี๋ยวนี้ ไม่งั้นจะสาย



You have to leave now or else you will be late

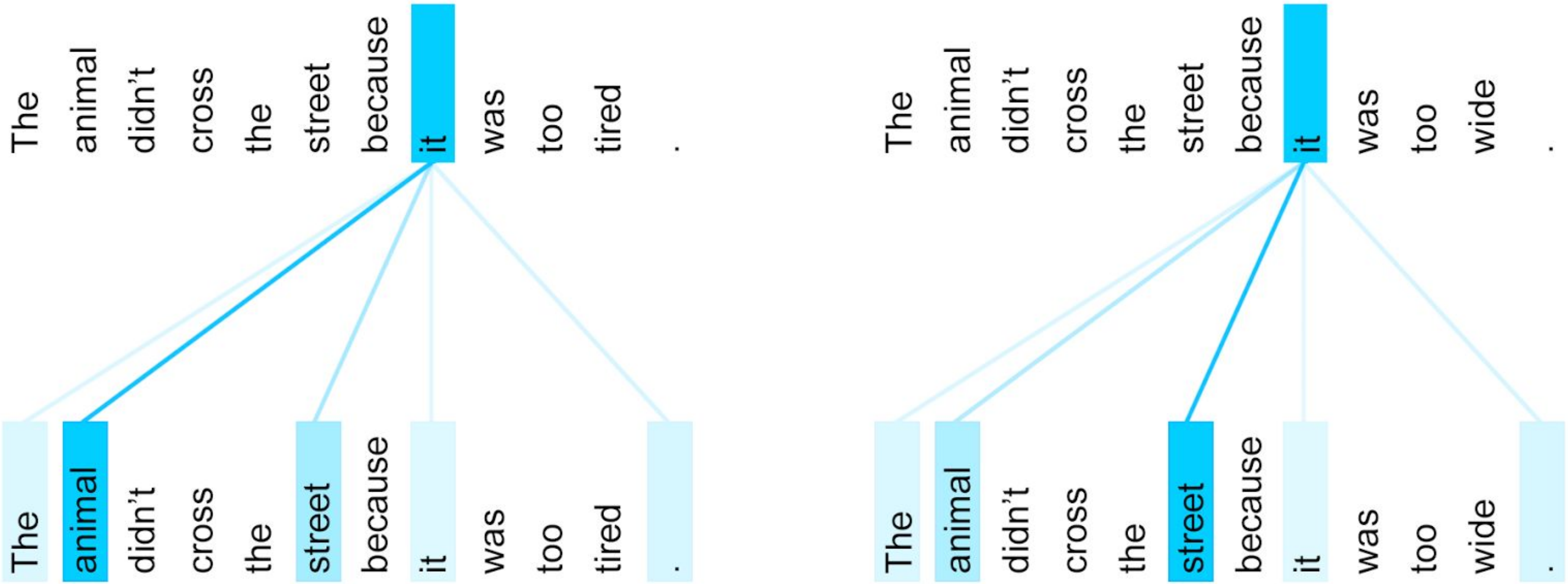
Attention


知
识
就
是
力
量
<end>
Encoder
e_0
e_1
e_2
e_3
e_4
e_5
e_6
Decoder
d_0
d_1
d_2
d_3
Knowledge
is
power
<end>

Attention

The animal didn't cross the street because it was too tired.
L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

The animal didn't cross the street because it was too wide.
L'animal n'a pas traversé la rue parce qu'elle était trop large.

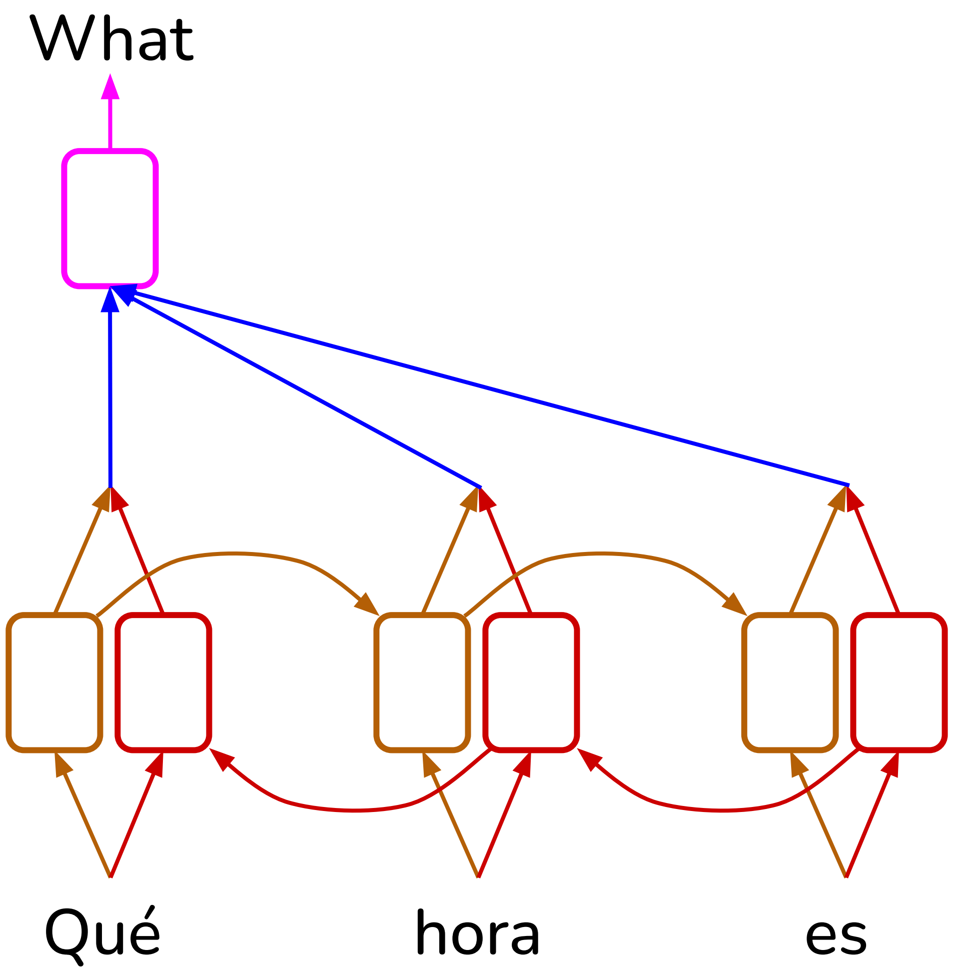


Attention

Decoder

Attention weights

Encoder

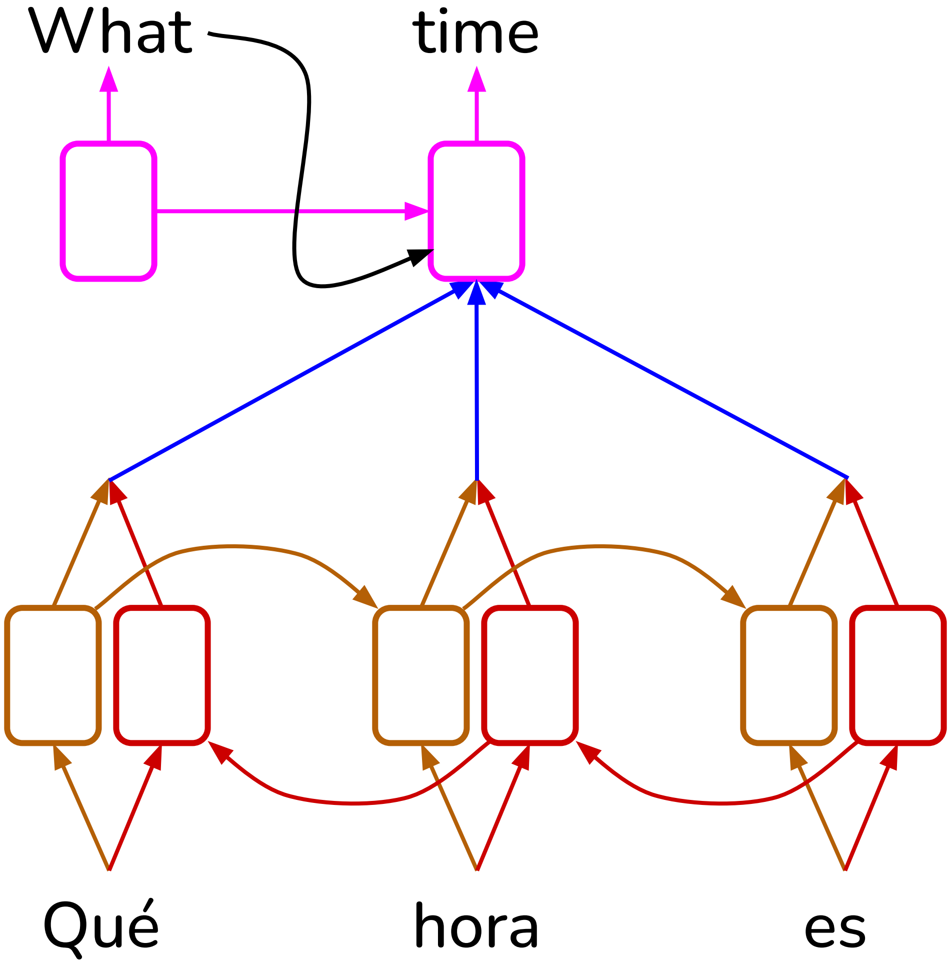


Attention

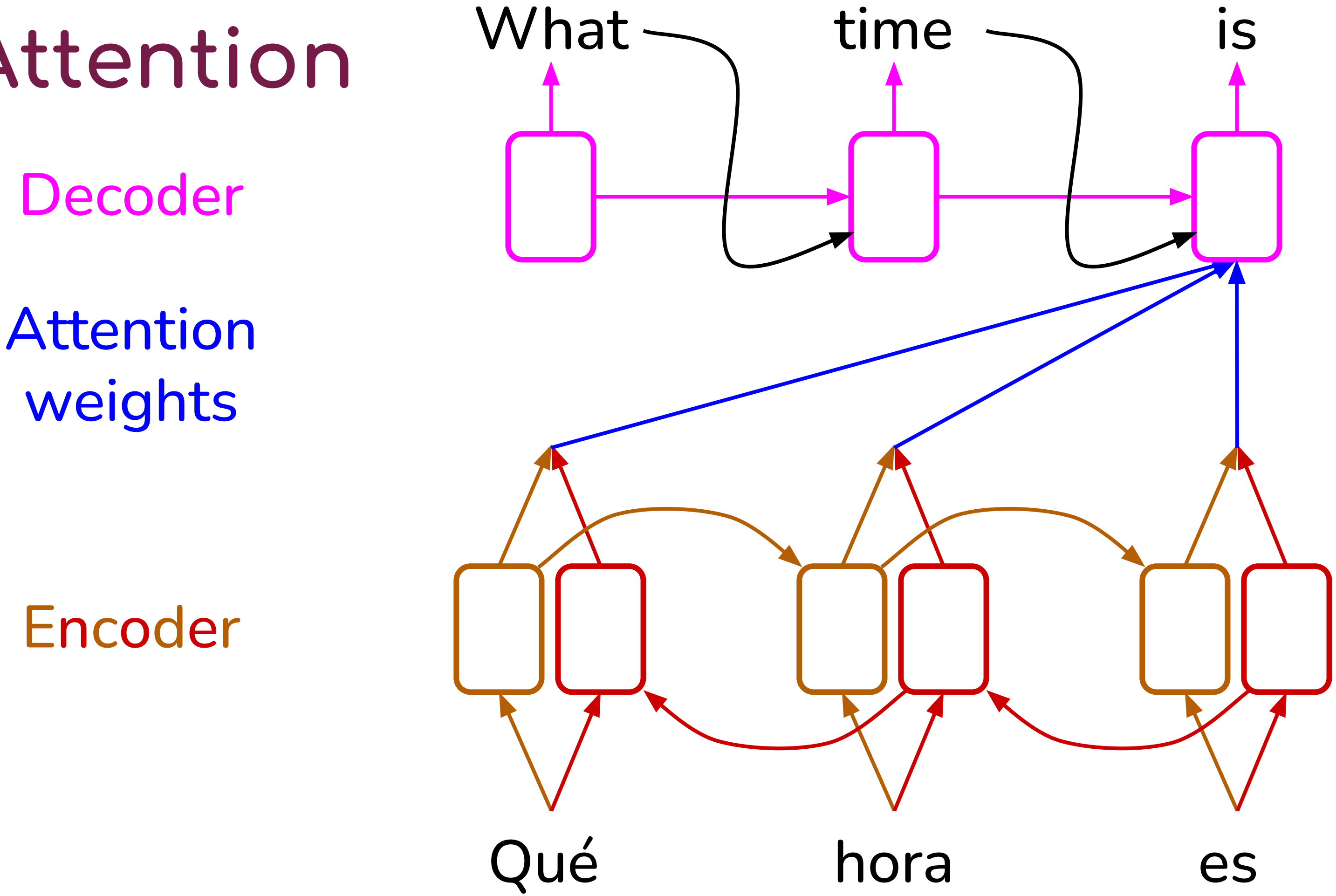
Decoder

Attention weights

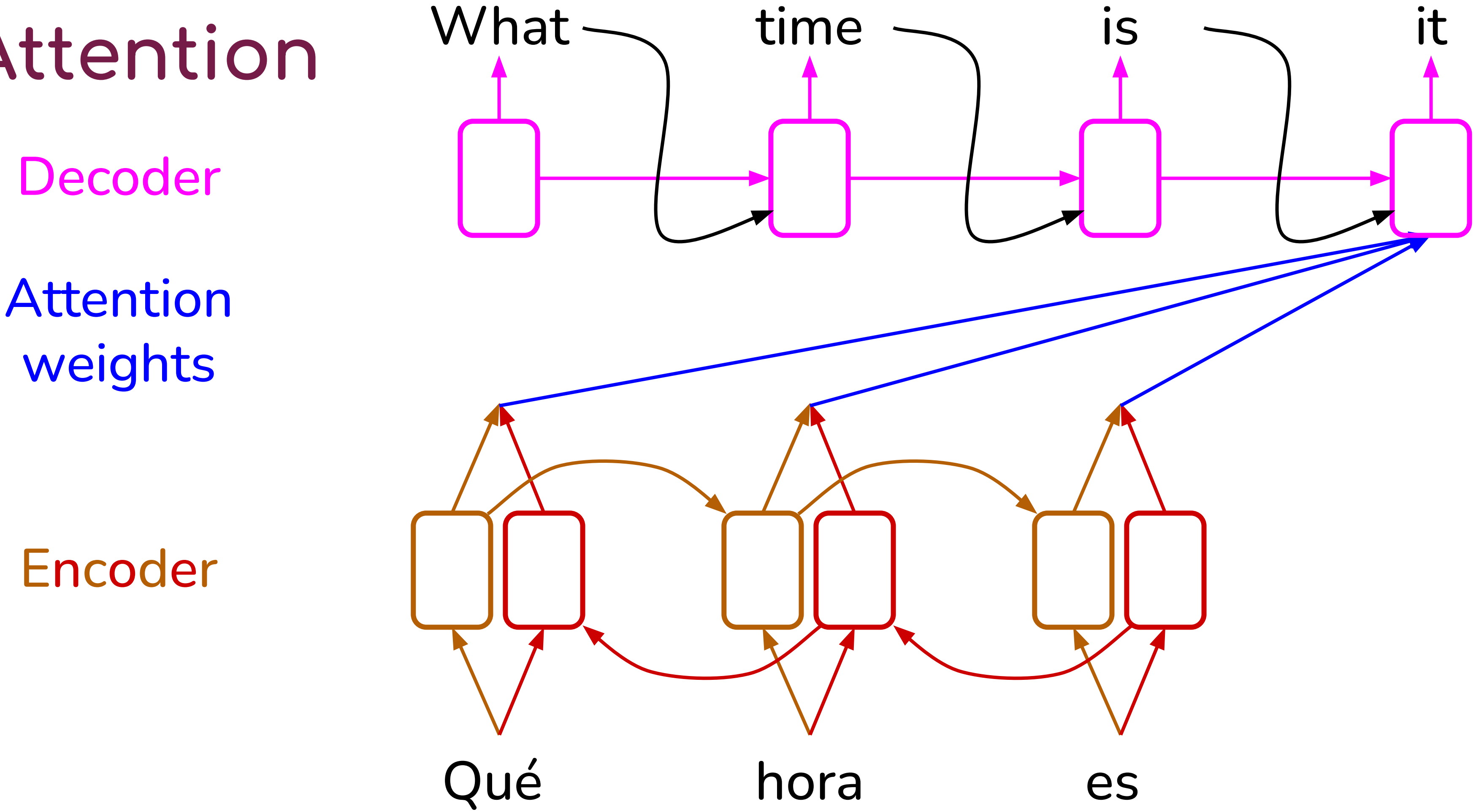
Encoder



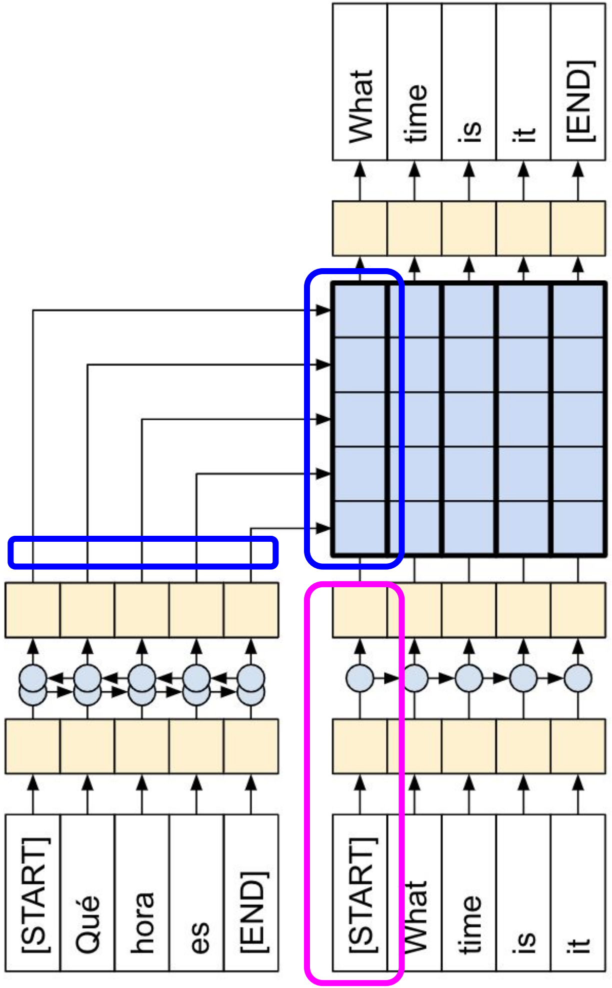
Attention



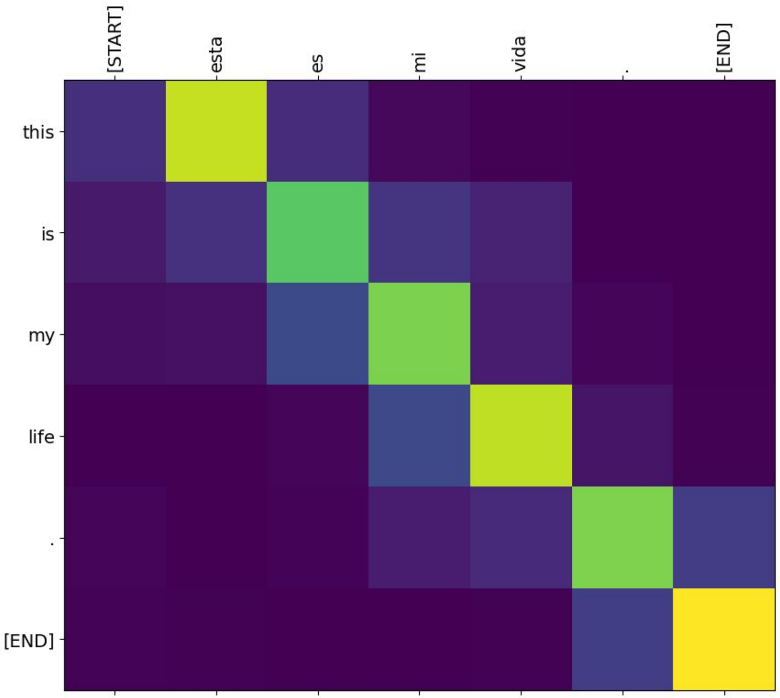
Attention

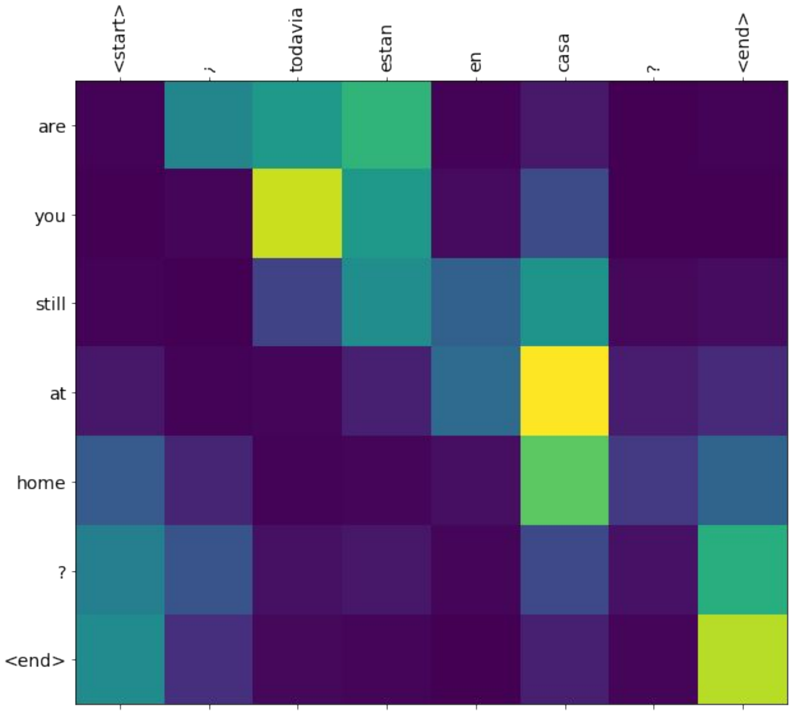

What
time
is
it
Decoder
Attention
weights
Encoder
Qué
hora
es

Attention



Heatmap of Attention Weights



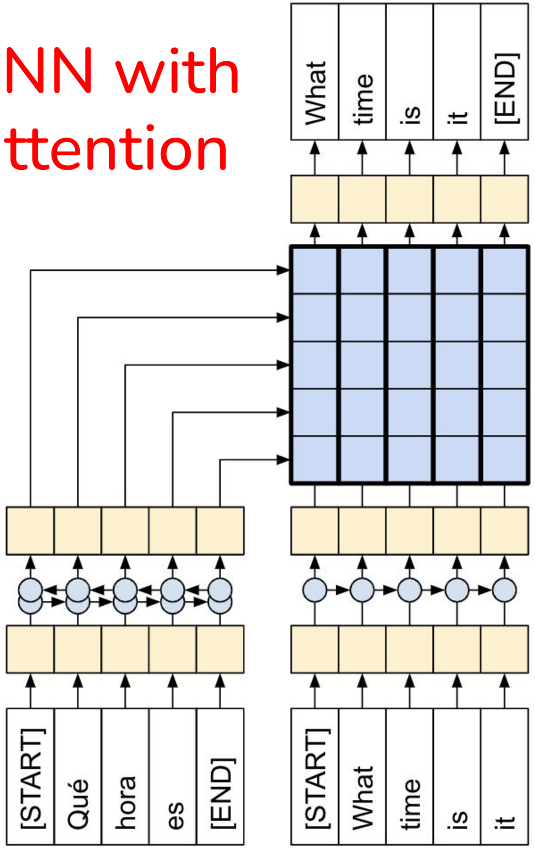


RNNs

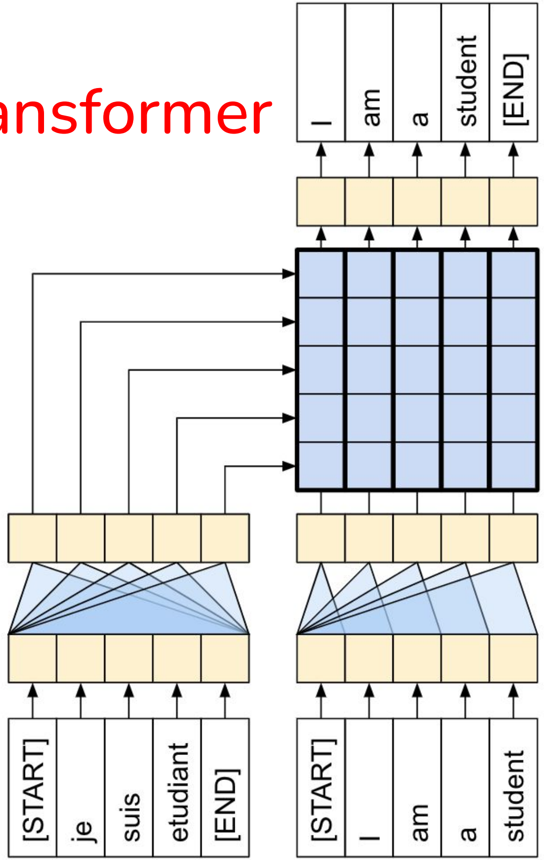
- Recap
- GRUs and LSTMs
- Bidirectional
- Attention
- Transformers

Transformers

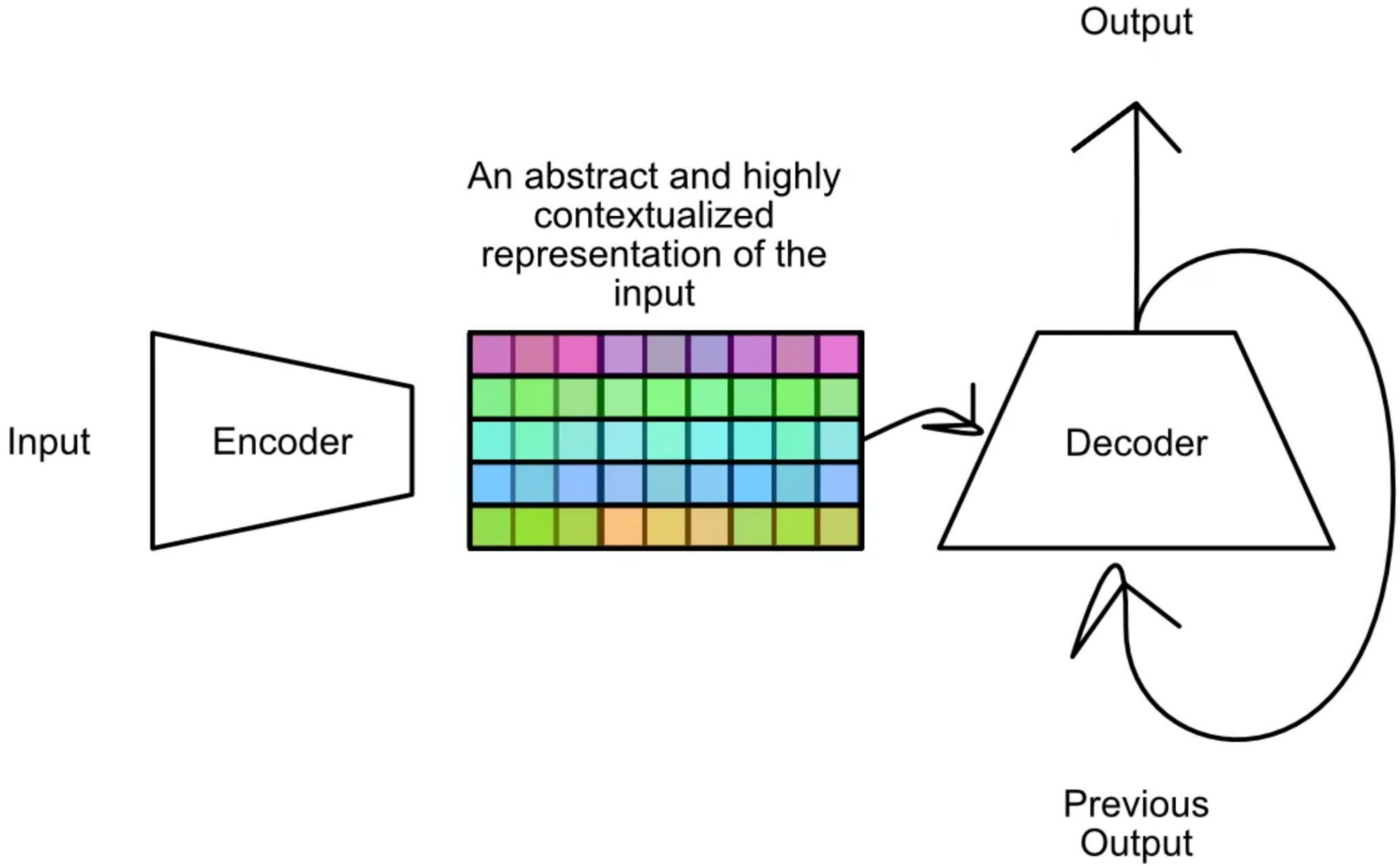
RNN with attention



Transformer



Transformers



Transformers

- *Self-attention*. Rather than use a fixed embedding for each word regardless of where/how it's used in a sentence, a context-dependent embedding is calculated for each word based on its relationship to other words in the sentence. Thus, a richer representation for each word.

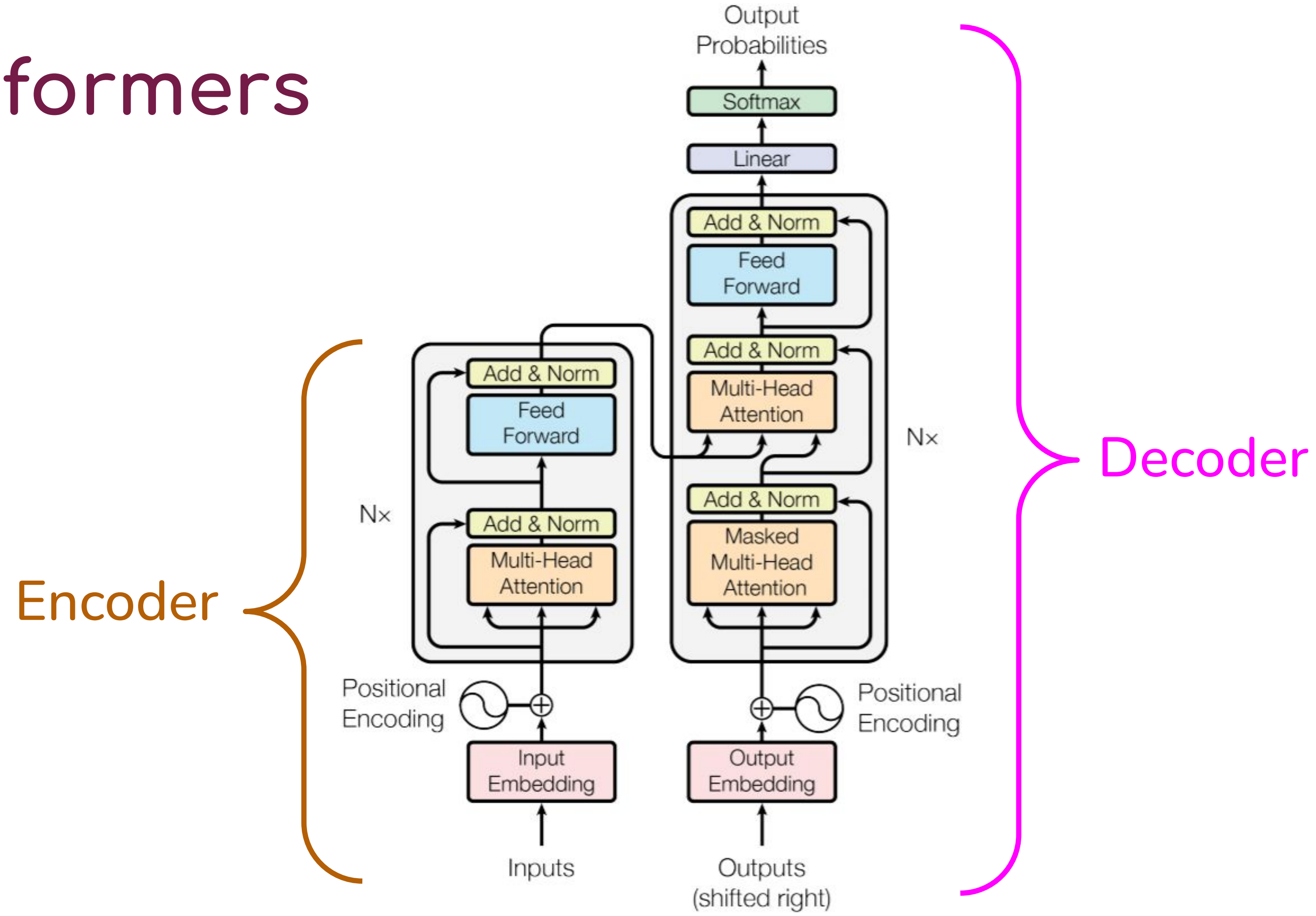
- *Multi-headed attention*. Multiple self-attention representations are calculated

- *Positional embedding*. The position of each word in a sequence is incorporated into the representation

- *Parallelization!*

Transformers

Transformers



Large Language Models (LLMs)

- Use decoder portion of transformer architecture
- Trillions of parameters, trained on terabytes of data with trillions of words
- Costs $10s or $100s of millions to train
- Carbon footprint - 500 tons of CO_2e for training and 1000 tons of CO_2e per month for inference