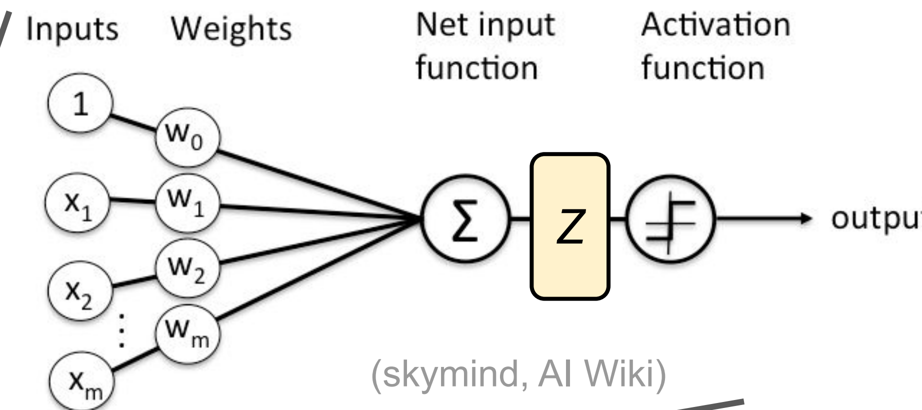
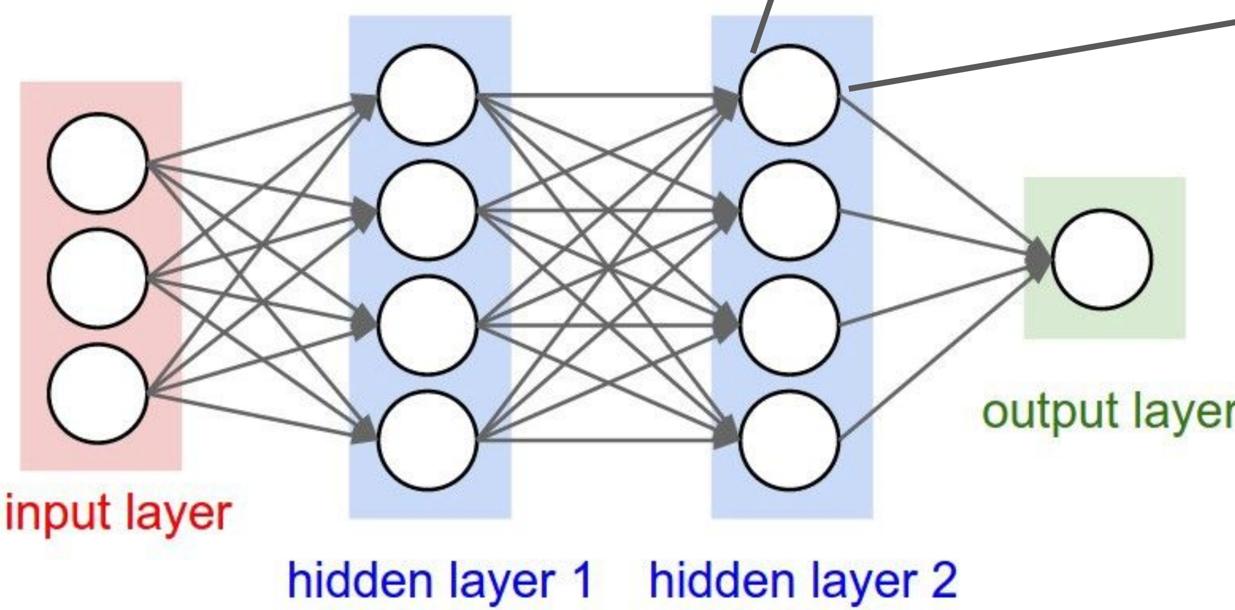
Transformer Models

CSE545 - Spring 2019

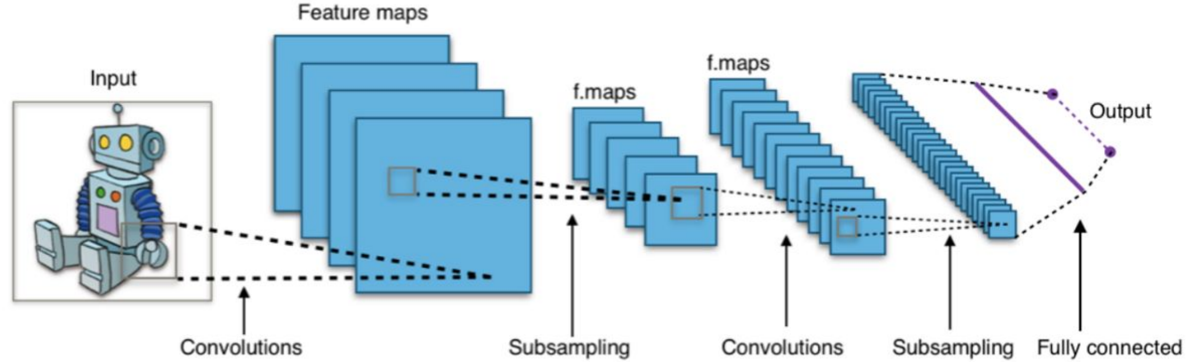
Review:
Feed Forward Network
(full-connected)

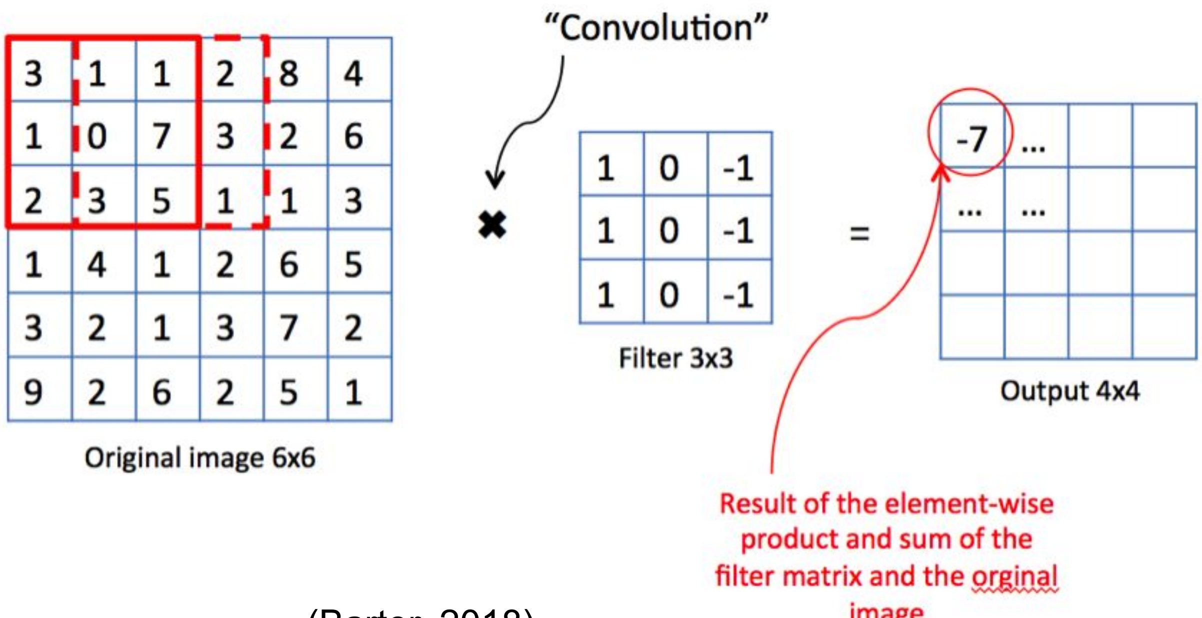


(skymind, AI Wiki)



Review: Convolutional NN





(Barter, 2018)

Review:
Recurrent Neural Network

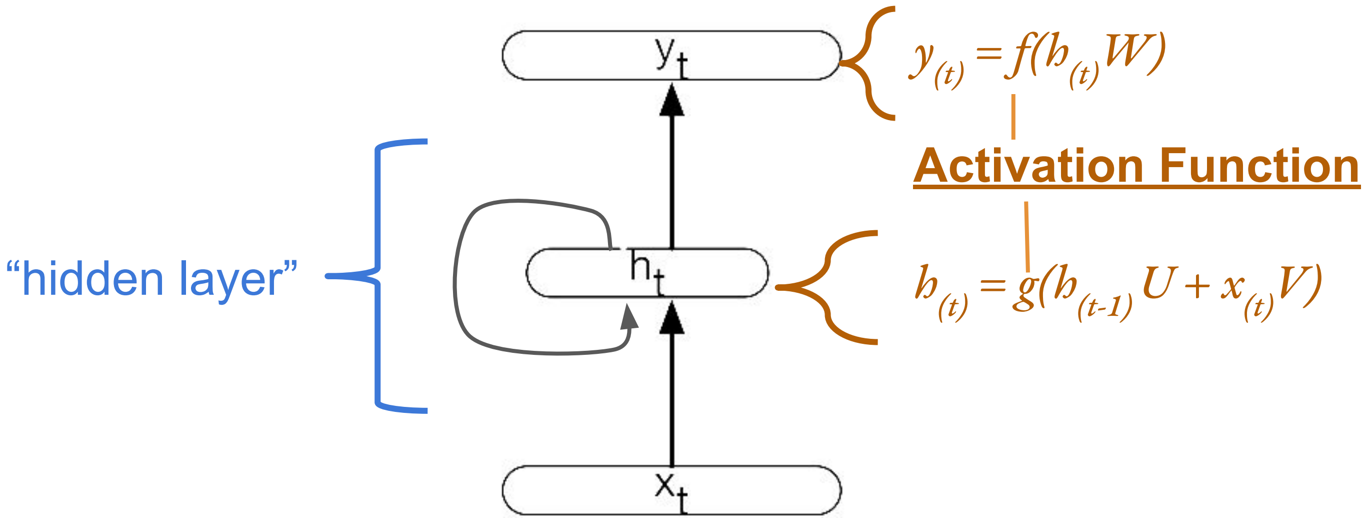
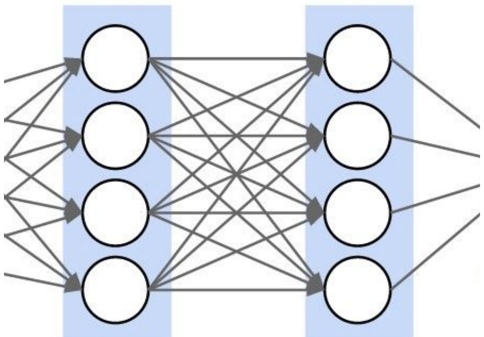
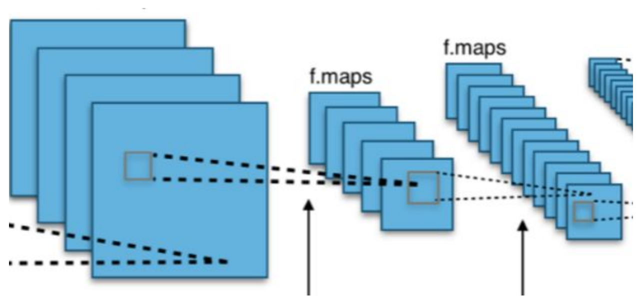


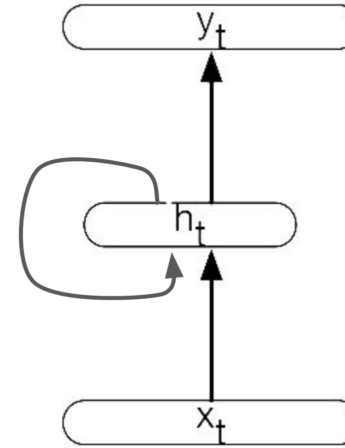
Figure 9.2 Simple recurrent neural network after Elman (Elman, 1990). The hidden layer includes a recurrent connection as part of its input. That is, the activation value of the hidden layer depends on the current input as well as the activation value of the hidden layer from the previous timestep.

(Jurafsky, 2019)

FFN | **CNN** | **RNN**

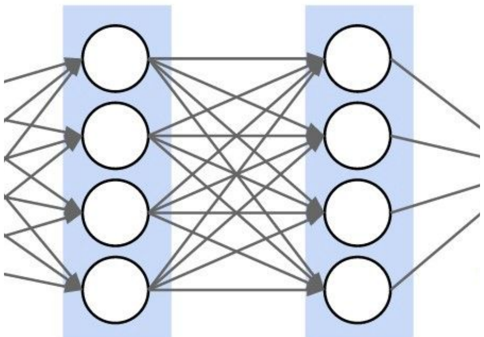


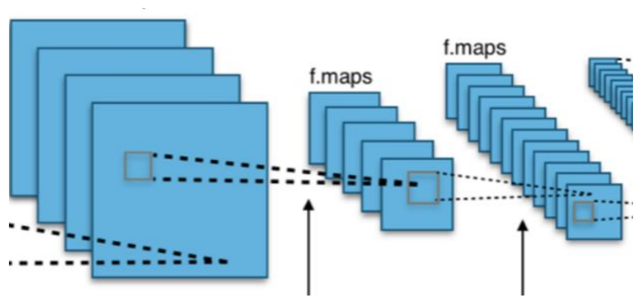


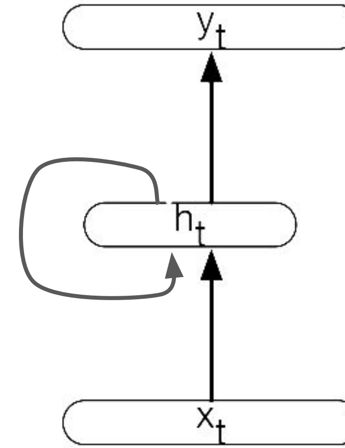


Can model computation (e.g. matrix operations for a single input) be parallelized?

FFN **CNN** **RNN**

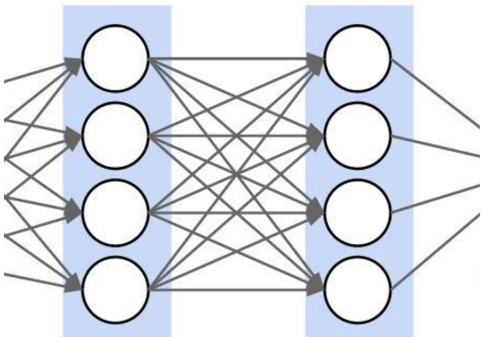


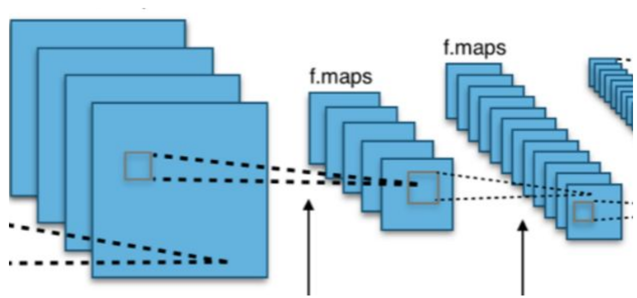


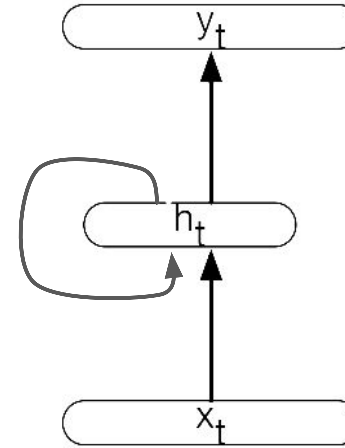


Can model computation (e.g. matrix operations for a single input) be parallelized?

FFN CNN RNN

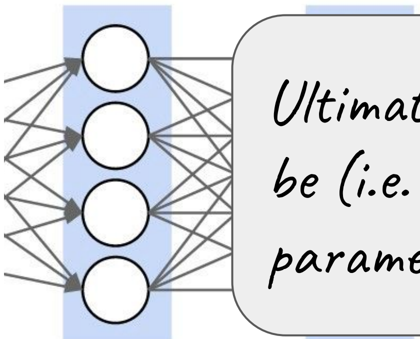


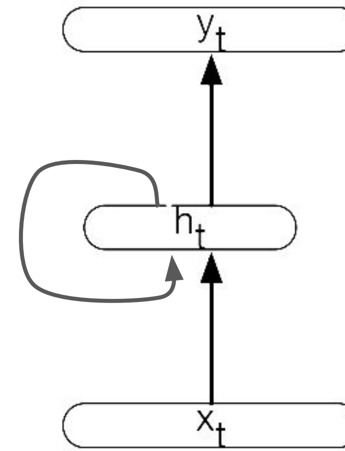




Can model computation (e.g. matrix operations for a single input) be parallelized?

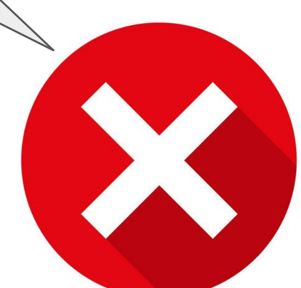
FFN **CNN** **RNN**





Ultimately limits how complex the model can be (i.e. it's total number of paramers/weights) as compared to a CNN.

Can model computation (e.g. matrix operations for a singl ... ut) be parallelized?



The Transformer: "Attention-only" models

Can handle sequences and long-distance dependencies, but….

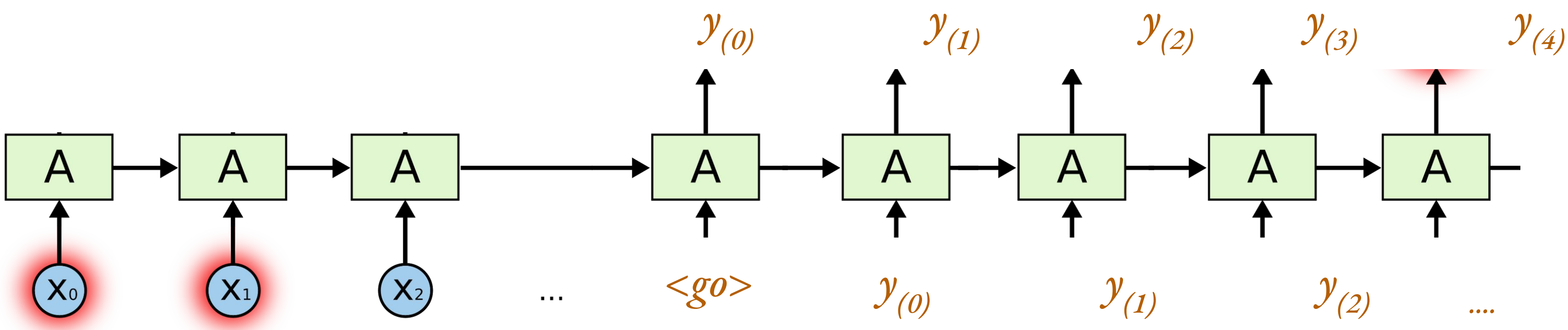
- Don't want complexity of LSTM/GRU cells
- Constant num edges between input steps
- Enables "interactions" (i.e. adaptations) between words
- **Easy to parallelize -- don't need sequential processing.**

The Transformer: “**Attention**-only” models

Challenge:

The ball was kicked by kayla.

- Long distance dependency when translating:



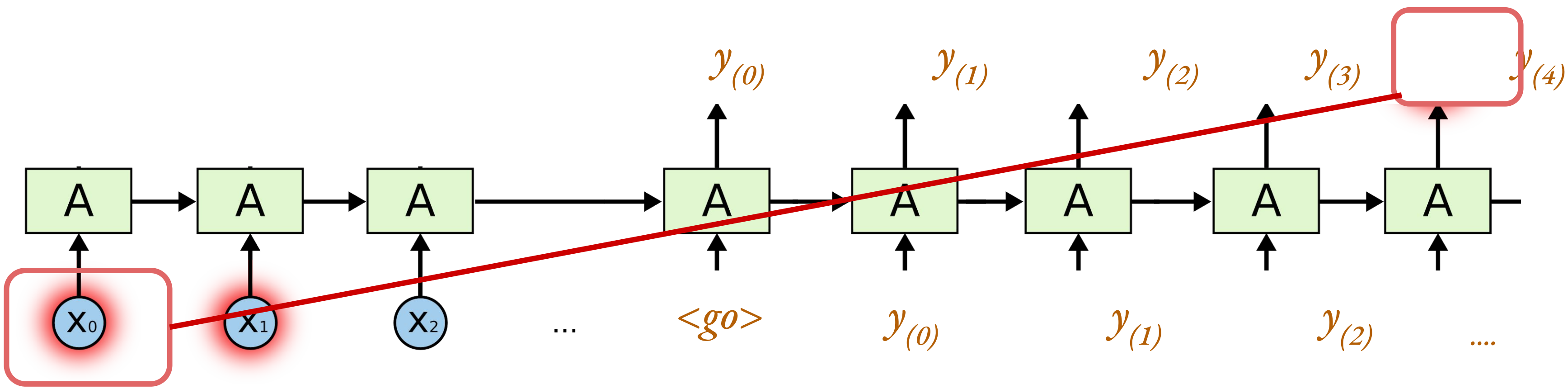
Kayla kicked the ball.

The Transformer: "**Attention**-only" models

Challenge:

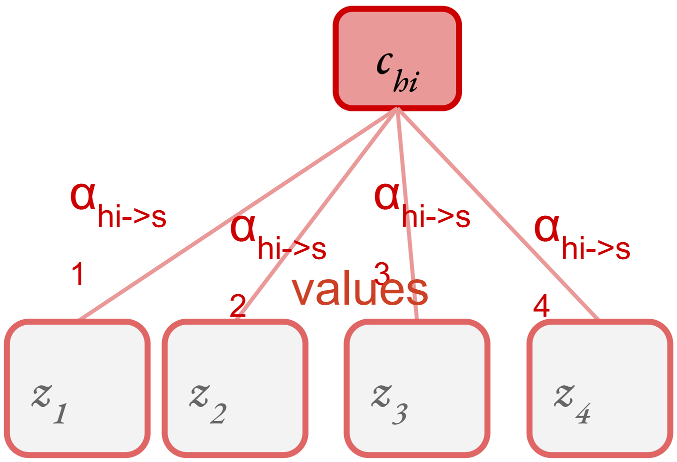
The ball was kicked by kayla.

- Long distance dependency when translating:

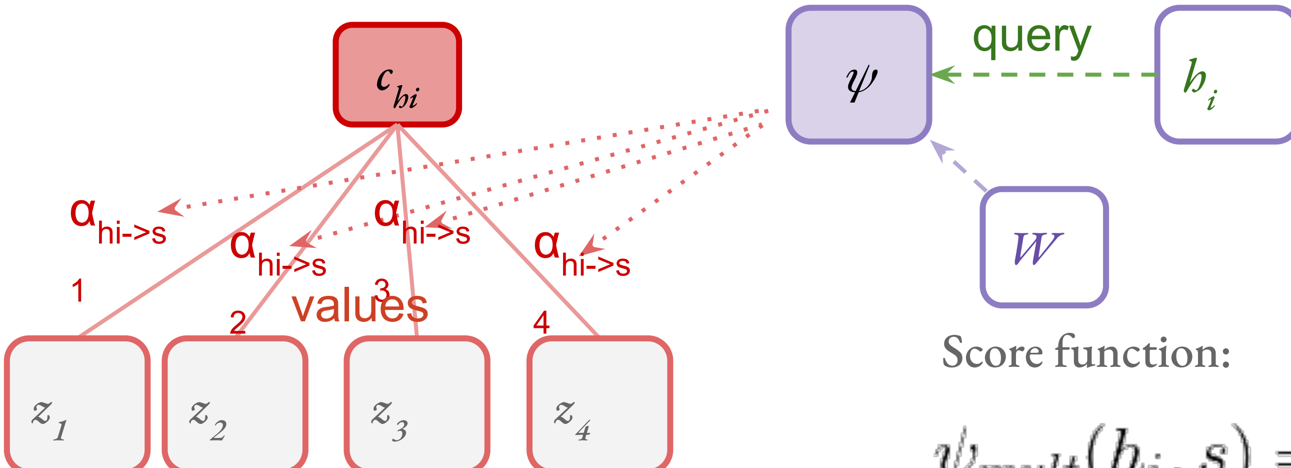


Kayla kicked the ball.

Attention


c_{hi}
$\alpha_{hi\text{->}s_1}$
$\alpha_{hi\text{->}s_2}$
$\alpha_{hi\text{->}s_3}$
$\alpha_{hi\text{->}s_4}$
values
z_1
z_2
z_3
z_4

Attention

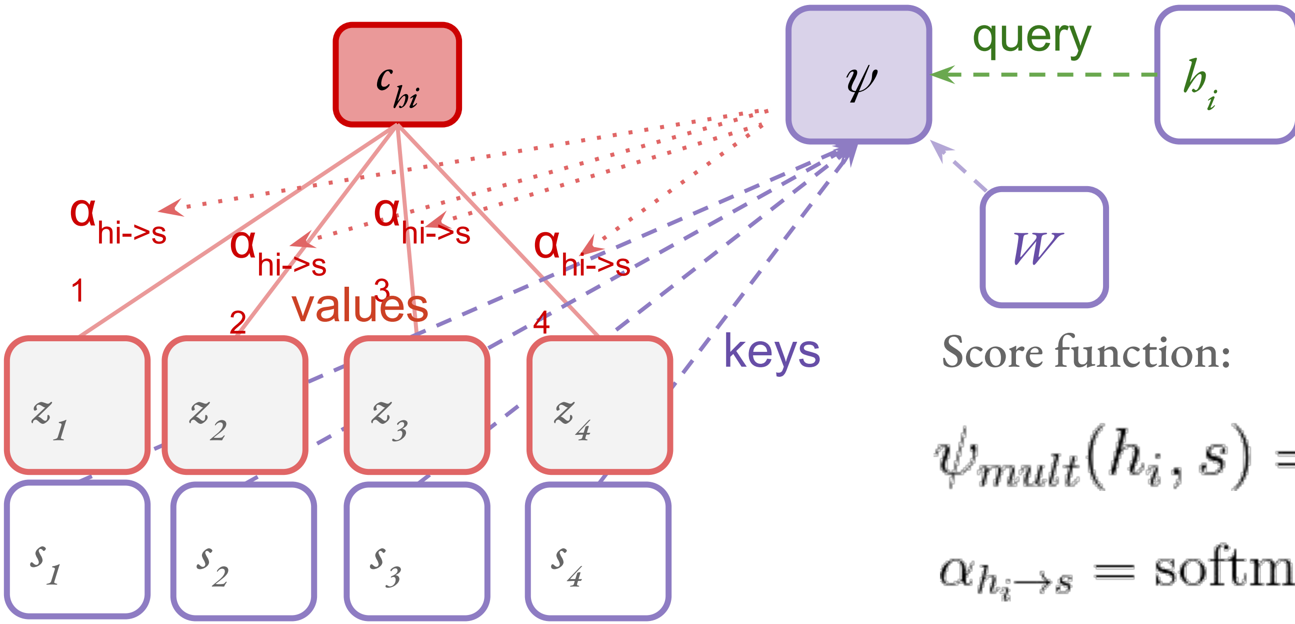


Score function:

$$\psi_{mult}(h_i, s) = s^T W h_i$$

$$\alpha_{h_i \rightarrow s} = \text{softmax}(\psi(h_i, s))$$

Attention



Score function:

$$\psi_{mult}(h_i, s) = s^T W h_i$$

$$\alpha_{h_i \to s} = \operatorname{softmax}(\psi(h_i, s))$$

$$c_{h_i} = \sum_{n=1}^{|s|} \alpha_{h_i \to s_n} z_n$$

The Transformer: “**Attention**-only” models

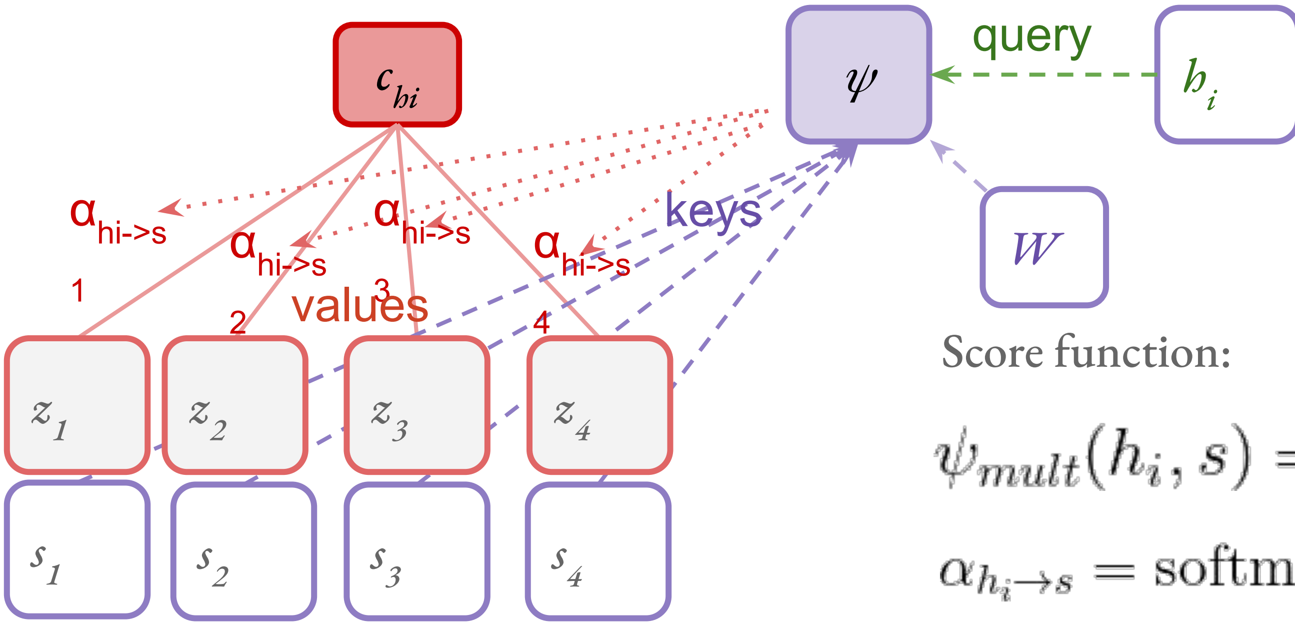
Challenge:

- Long distance dependency when translating:

Attention came about for encoder decoder models.

Then self-attention was introduced:

Attention



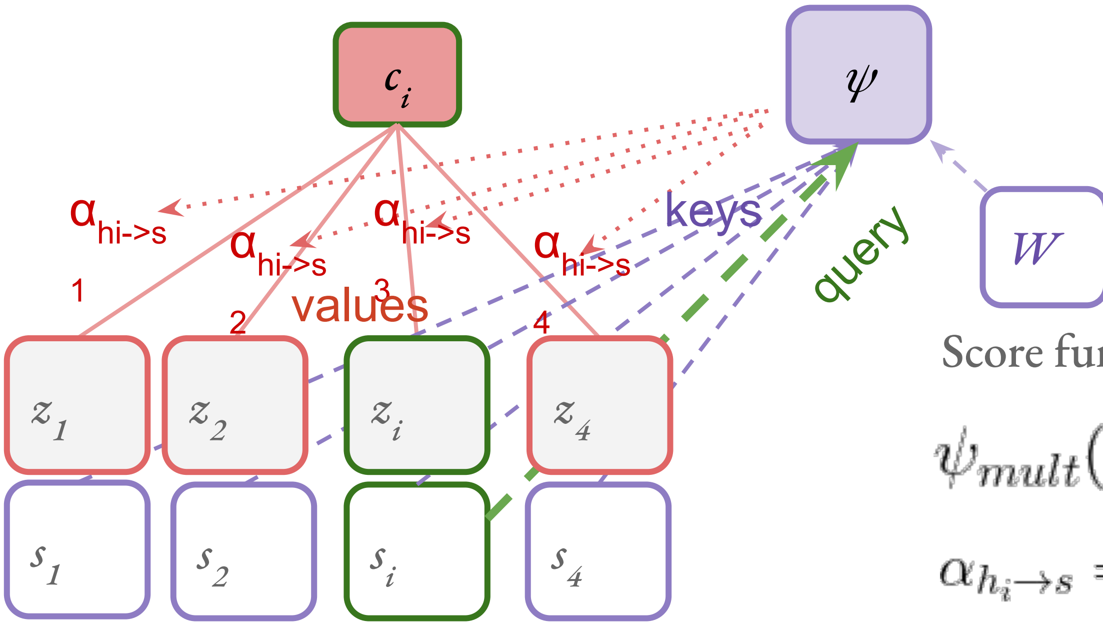
Score function:

$$\psi_{mult}(h_i, s) = s^T W h_i$$

$$\alpha_{h_i \to s} = \text{softmax}(\psi(h_i, s))$$

$$c_{h_i} = \sum_{n=1}^{|s|} \alpha_{h_i \to s_n} z_n$$

Self-Attention



Score function:

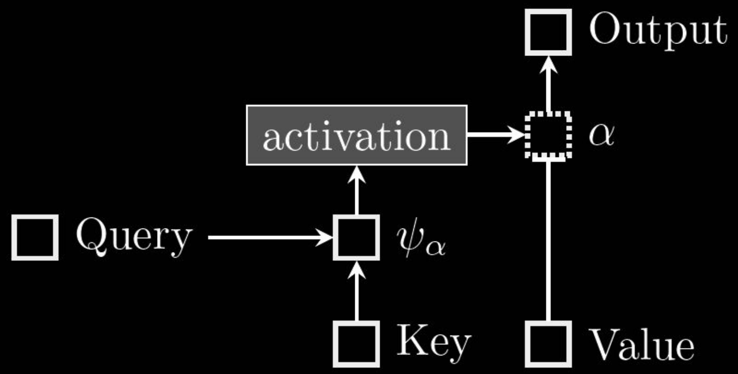
$$\psi_{mult}(h_i, s) = s^T W h_i$$

$$\alpha_{h_i \rightarrow s} = \mathrm{softmax}(\psi(h_i, s))$$

$$c_{h_i} = \sum_{n=1}^{|s|} \alpha_{h_i \rightarrow s_n} z_n$$

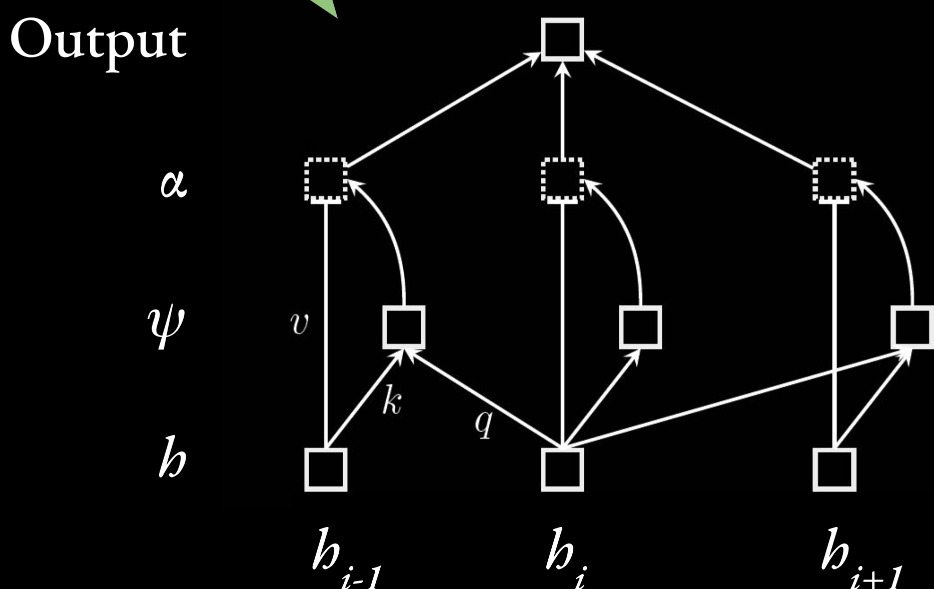
The Transformer: "Attention-only" models

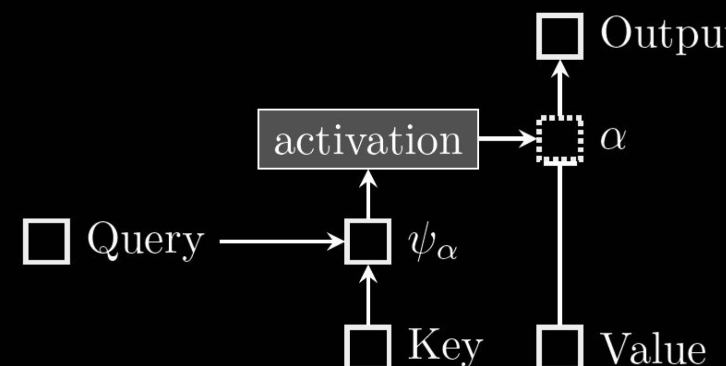
Attention as weighting a value based on a query and key:



(Eisenstein, 2018)

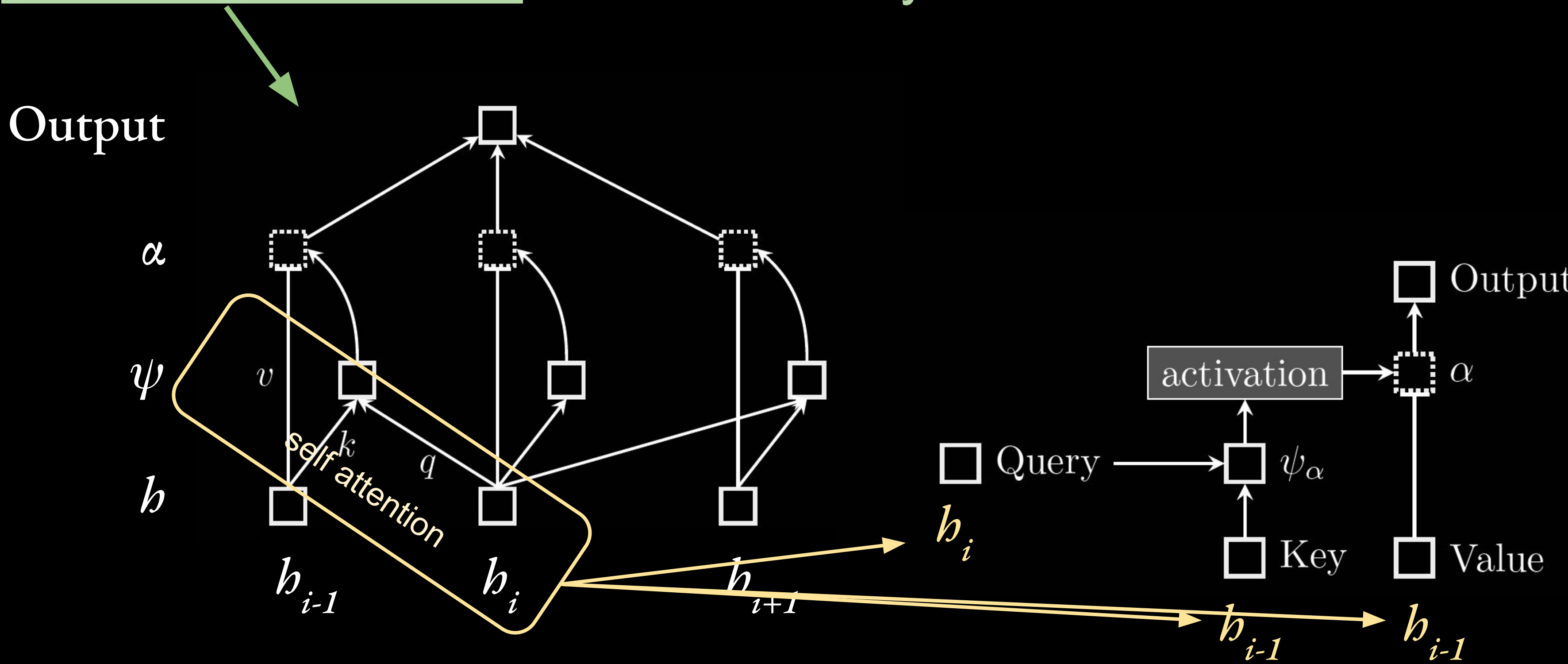
The Transformer: "Attention-only" models





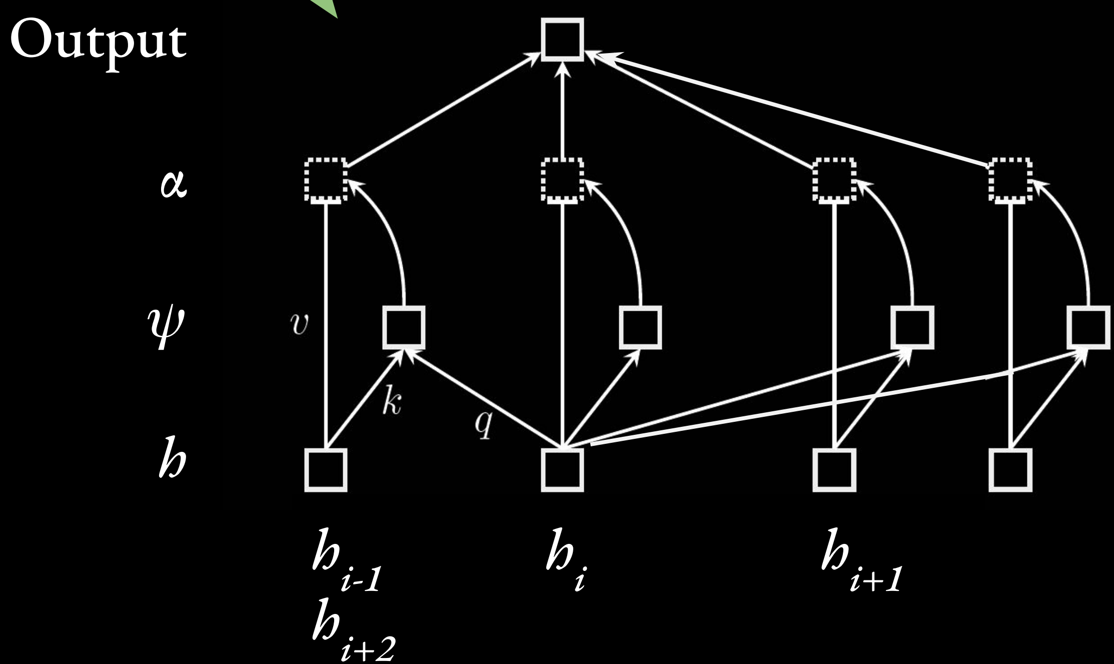
(Eisenstein, 2018)

The Transformer: "Attention-only" models

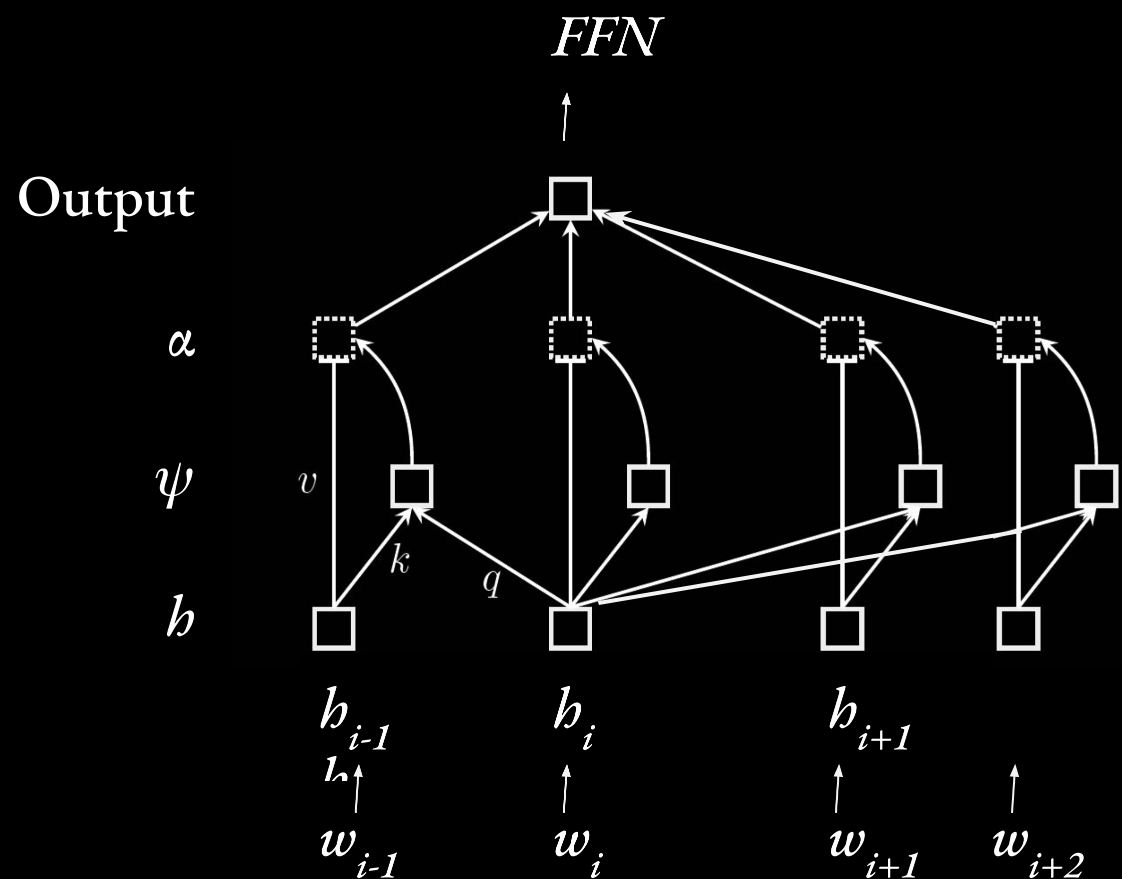


(Eisenstein, 2018)

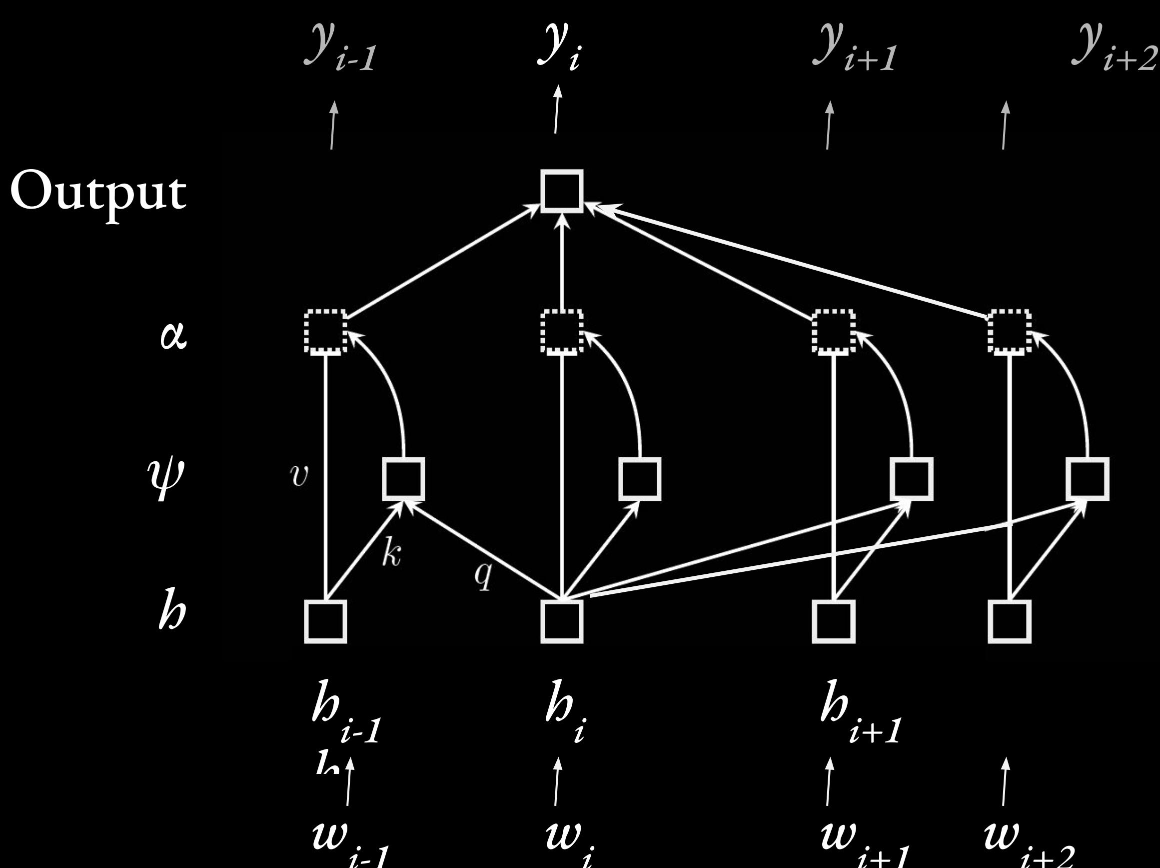
The Transformer: "Attention-only" models



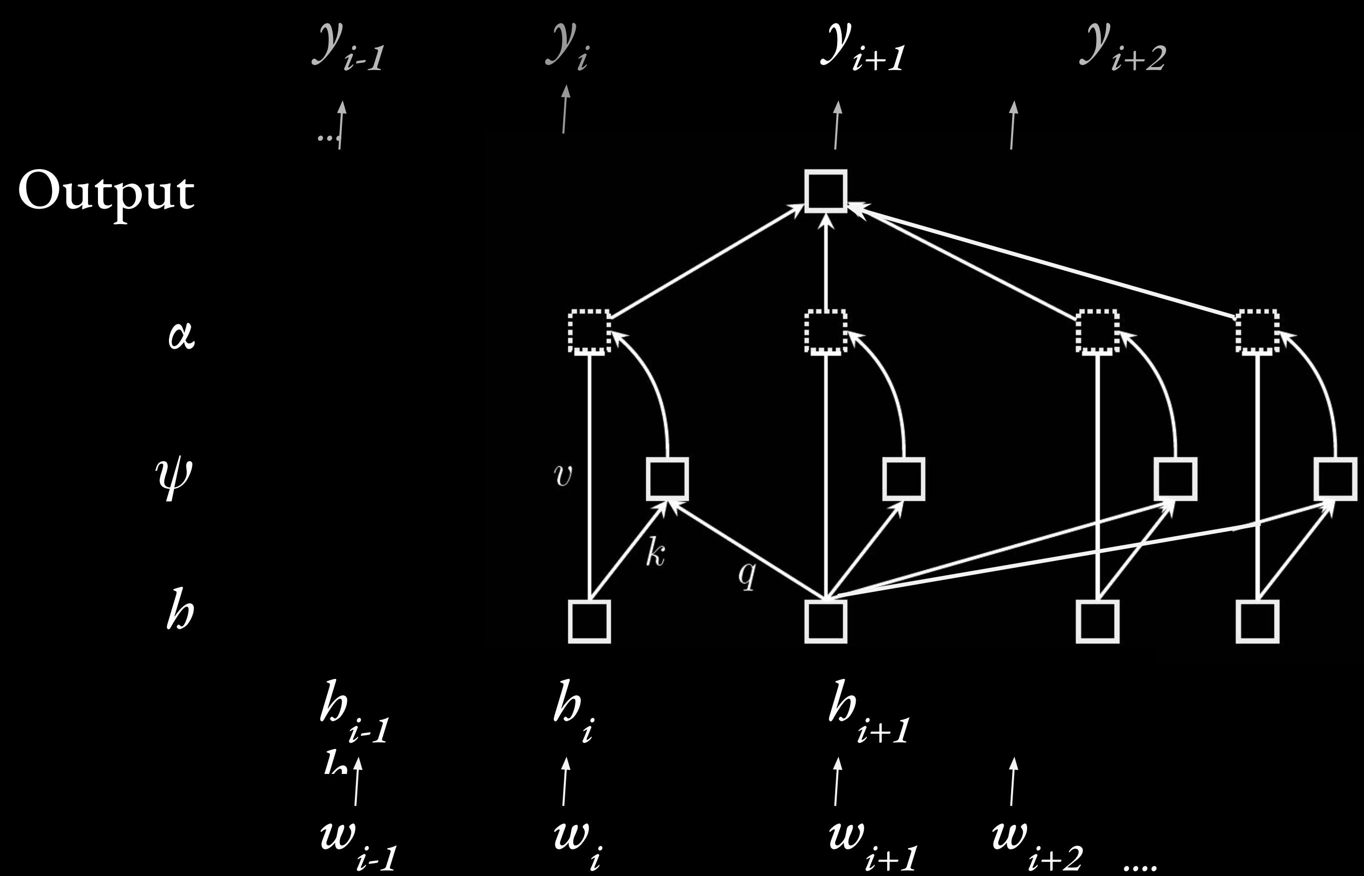
The Transformer: “Attention-only” models



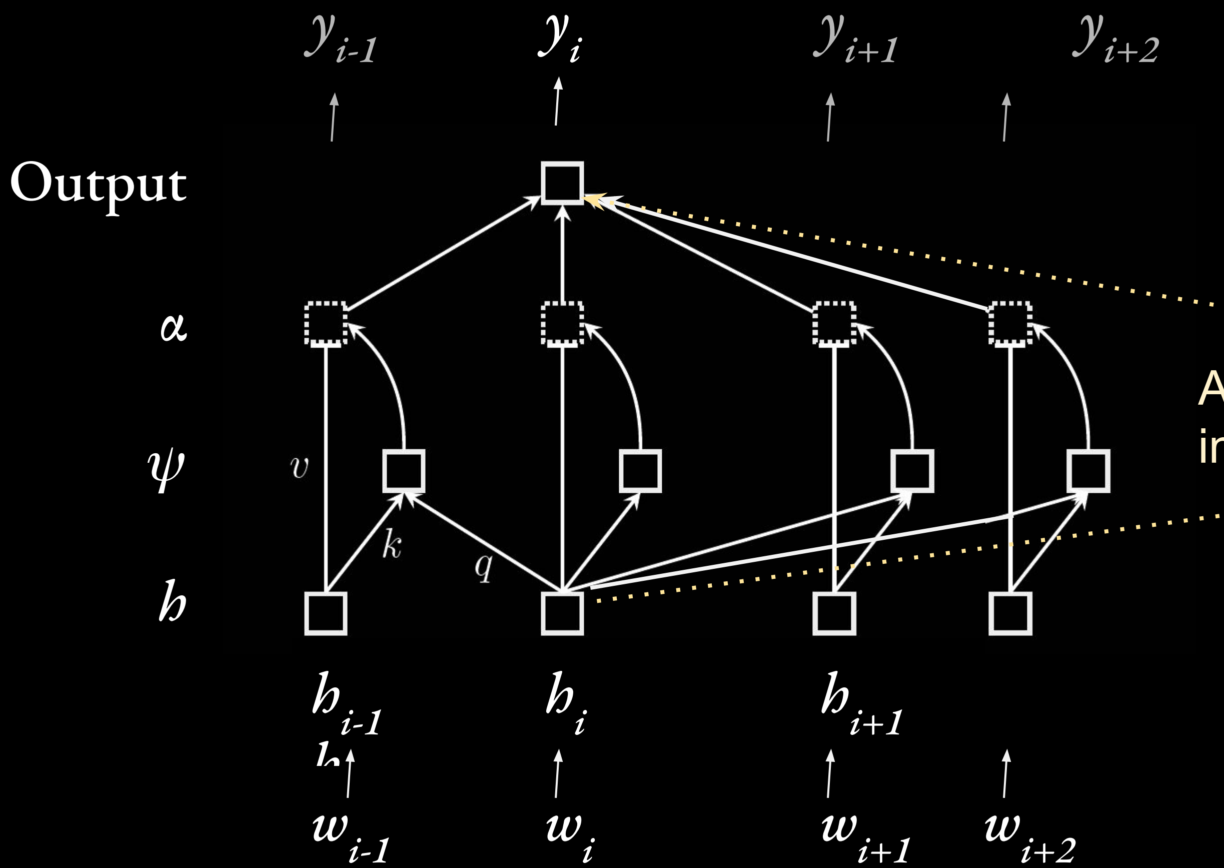
The Transformer: "Attention-only" models



The Transformer: “Attention-only” models

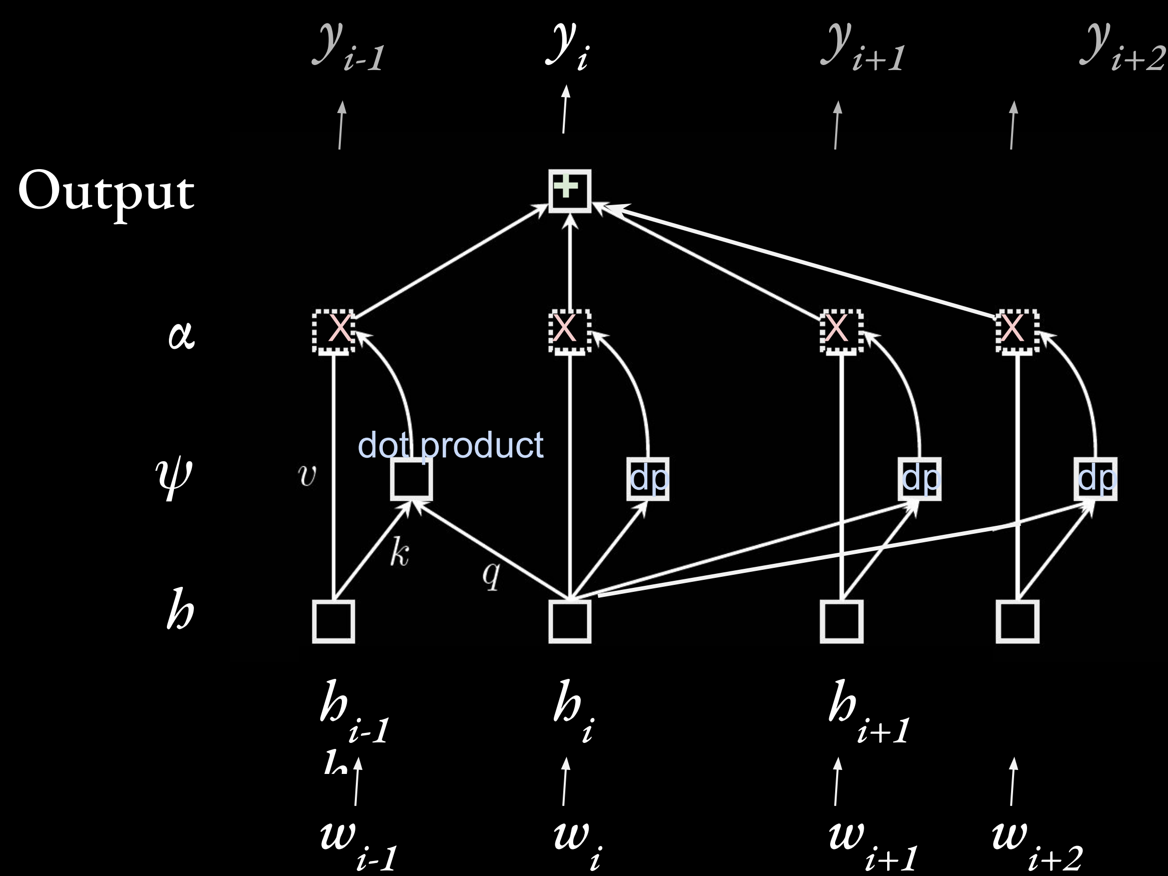


The Transformer: "Attention-only" models



Attend to all hidden states in your "neighborhood".

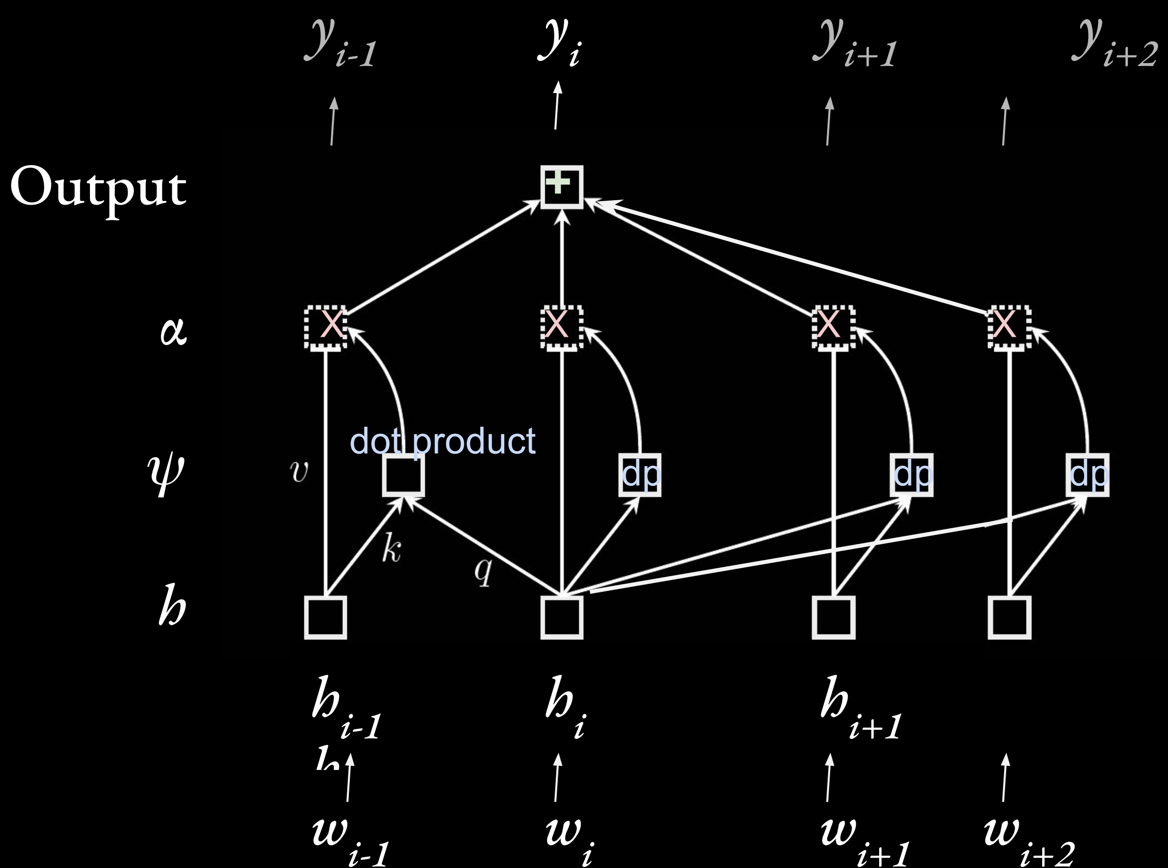
The Transformer: "Attention-only" models



$$\psi_{dp}(h_i, s) = s^T h_i$$

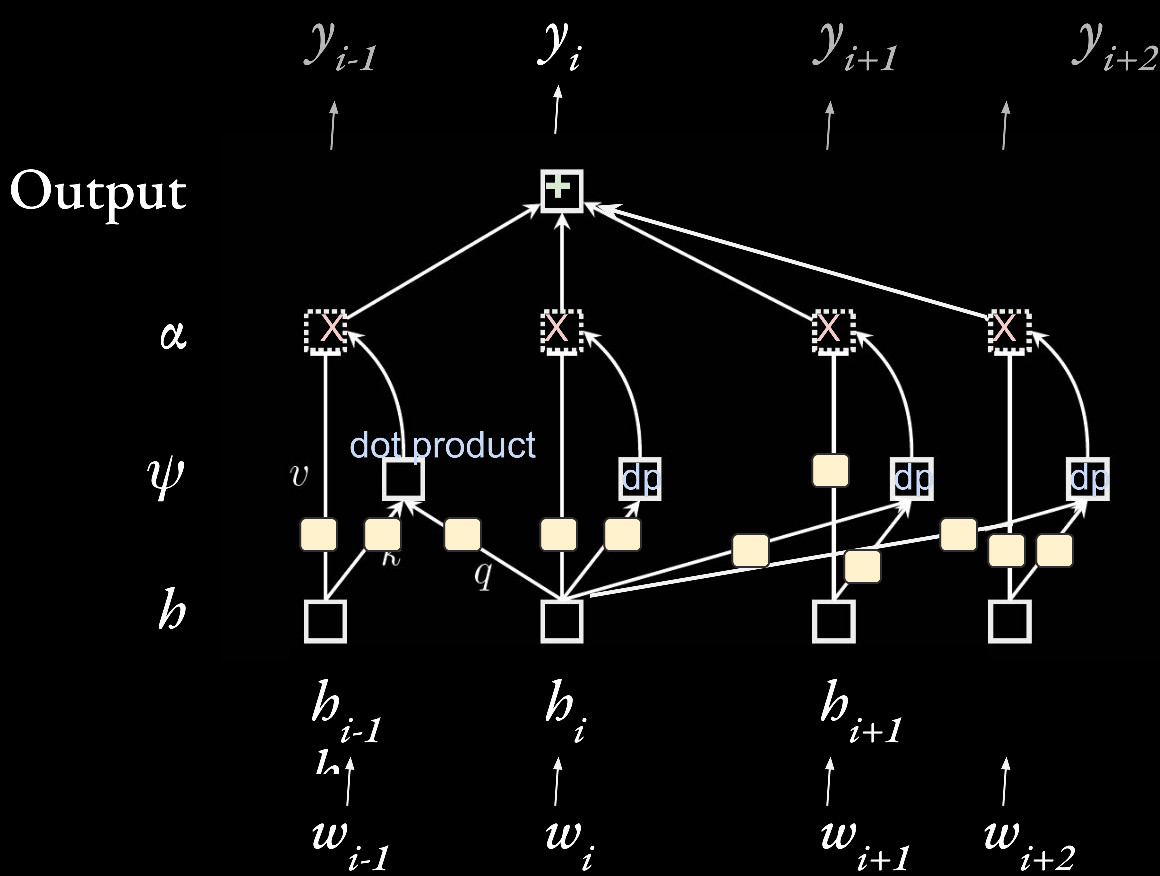
$k^t q$

The Transformer: "Attention-only" models



$$\psi_{dp}(k,q) = (k^t q)\,\overbrace{\sigma}^{\text{scaling parameter}}$$

The Transformer: “Attention-only” models



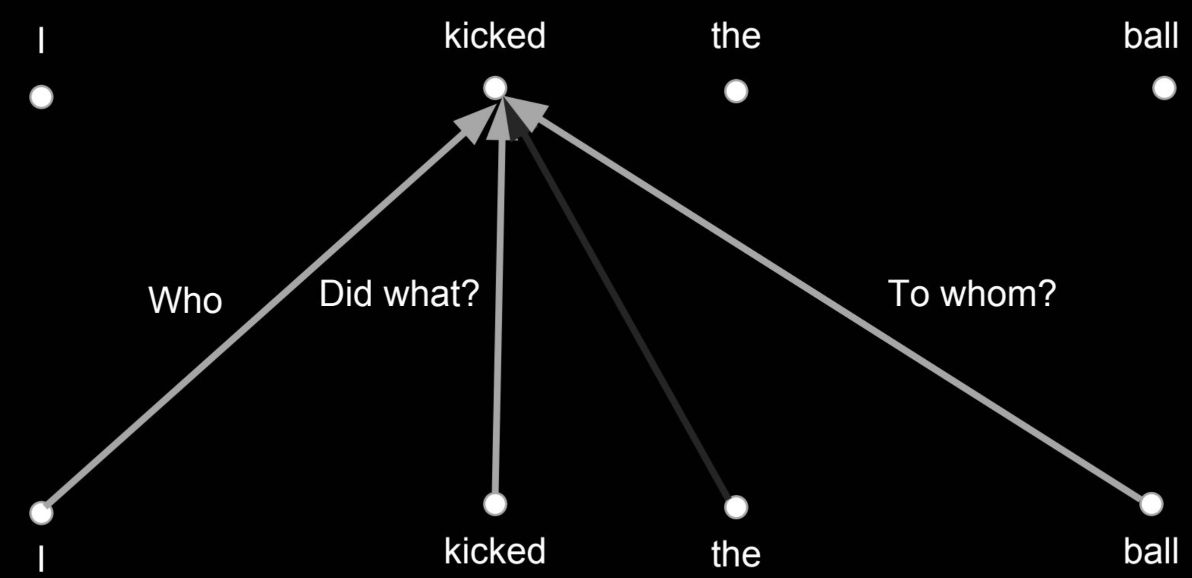
$$\psi_{dp}(k,q) = (k^t q)\,\sigma$$

Linear layer:
$W^T X$

One set of weights for each of for K, Q, and V

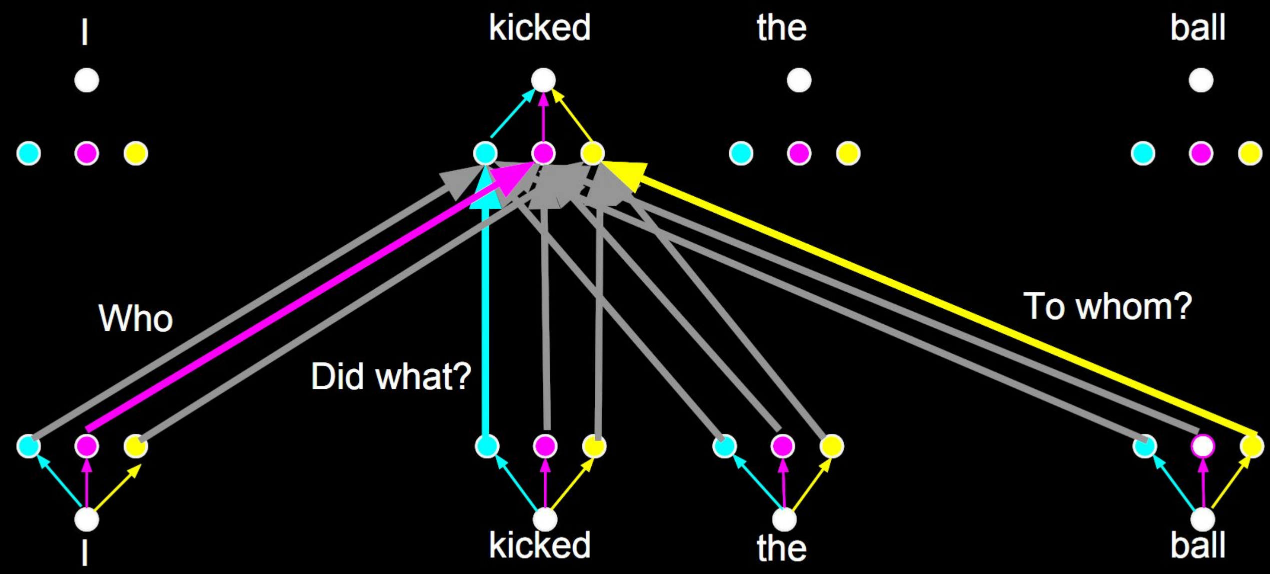
The Transformer

Limitation (thus far): Can’t capture multiple types of dependencies between words.

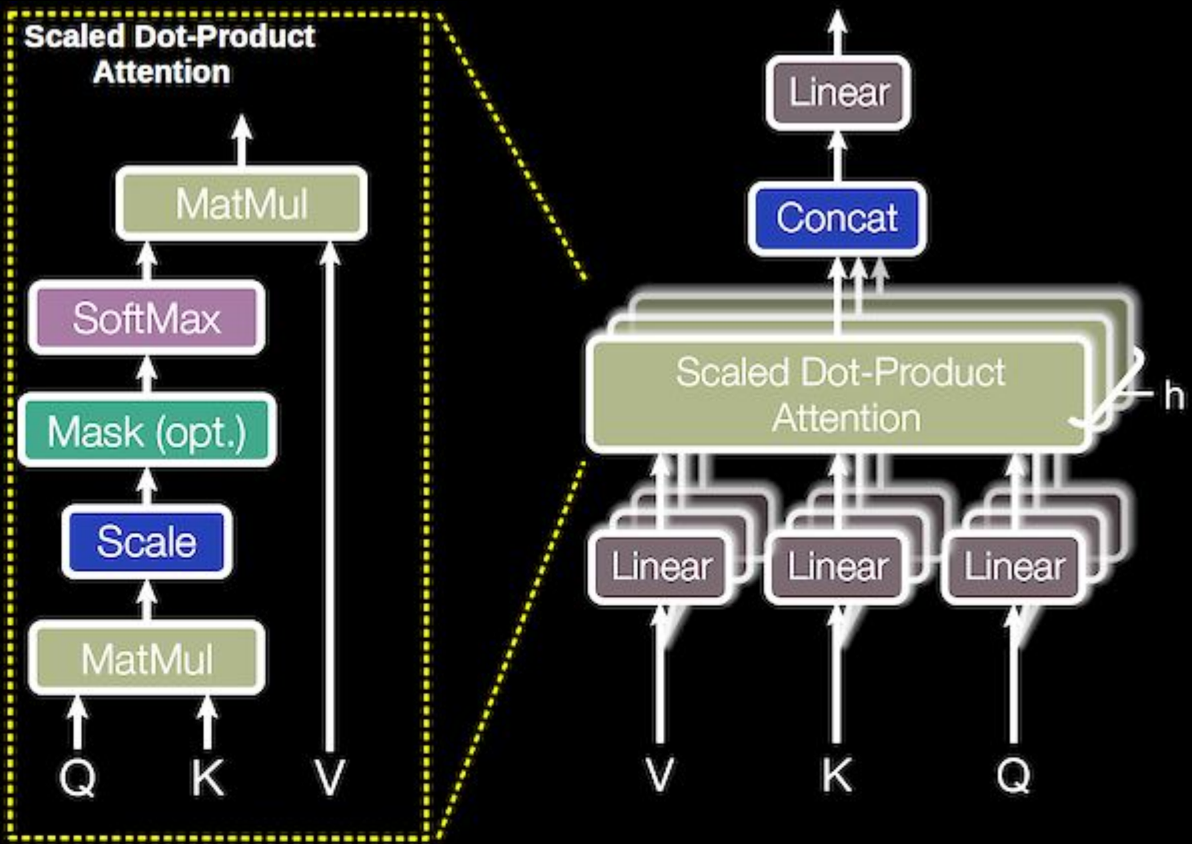


The Transformer

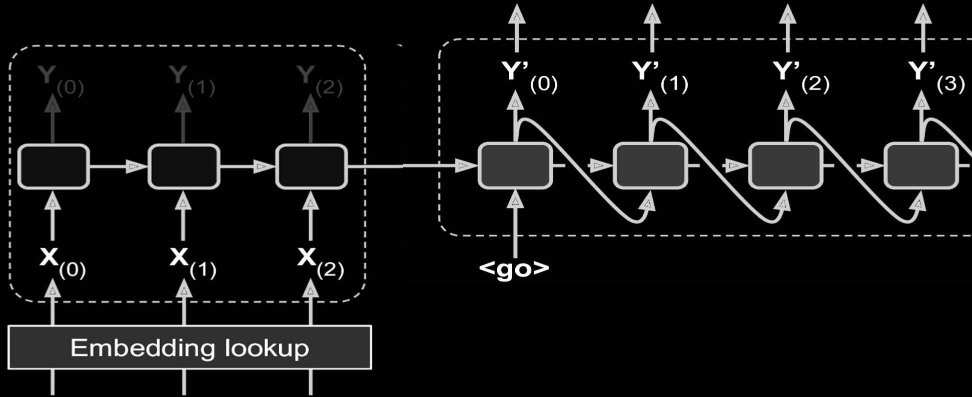
Solution: Multi-head attention

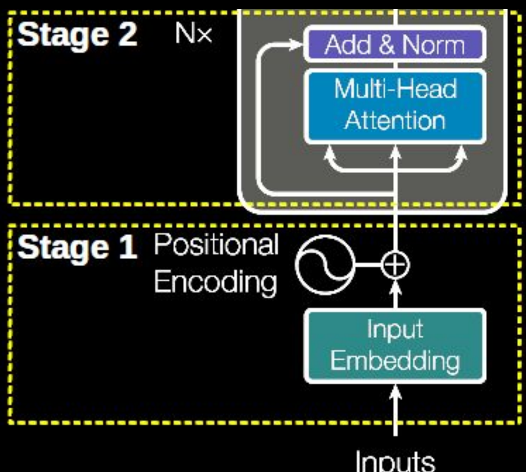


Multi-head Attention

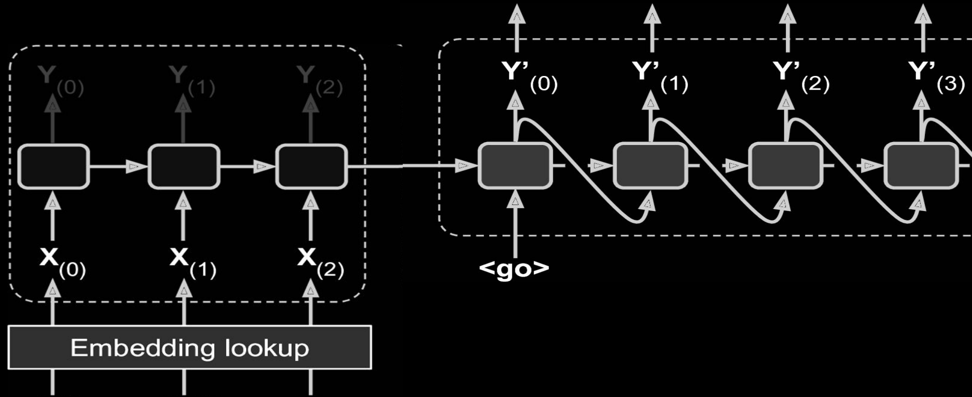


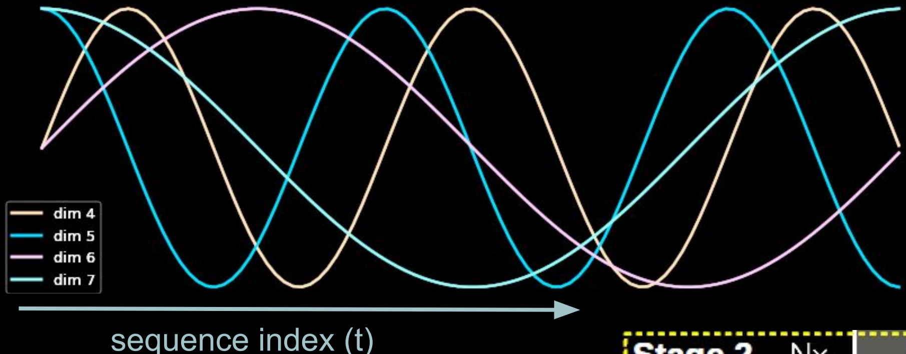
Transformer for Encoder-Decoder

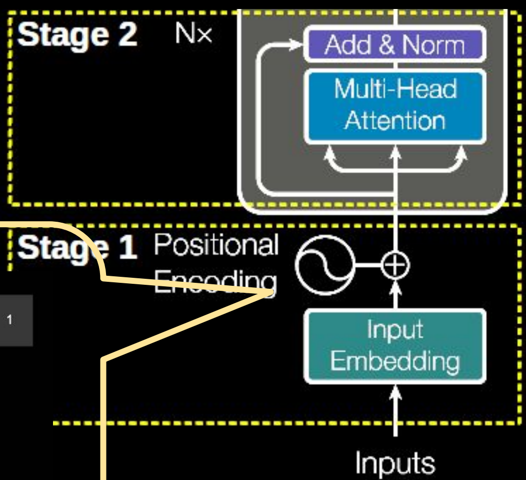


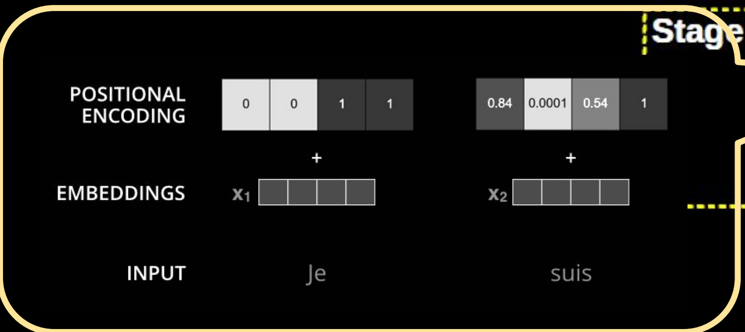


Transformer for Encoder-Decoder

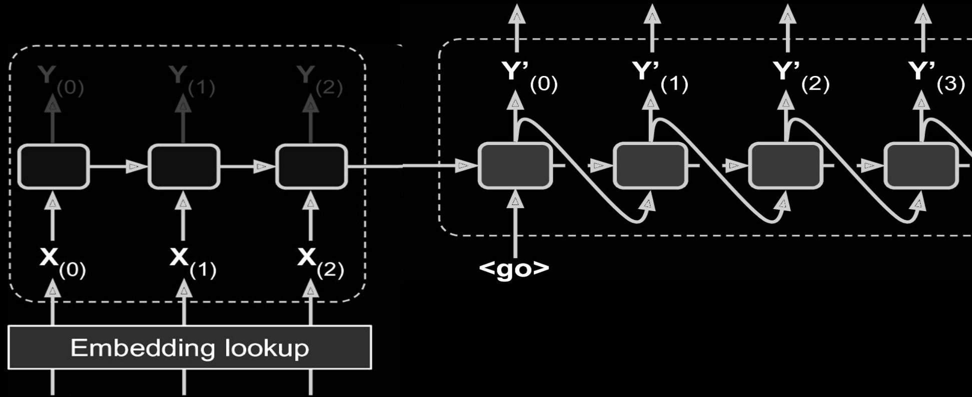


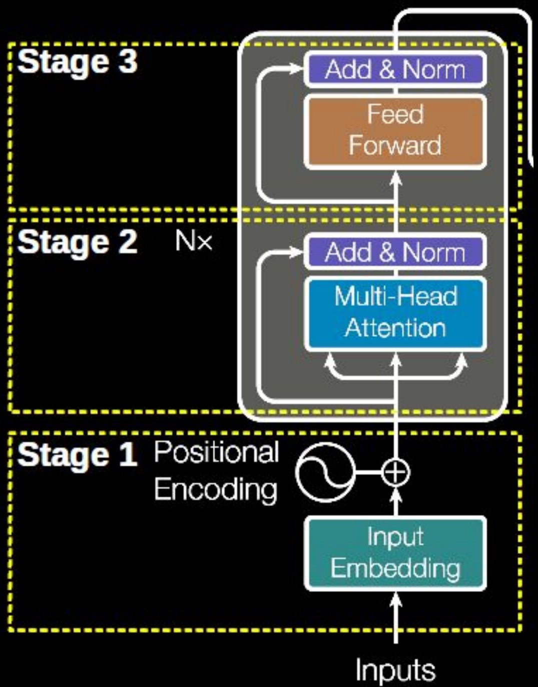




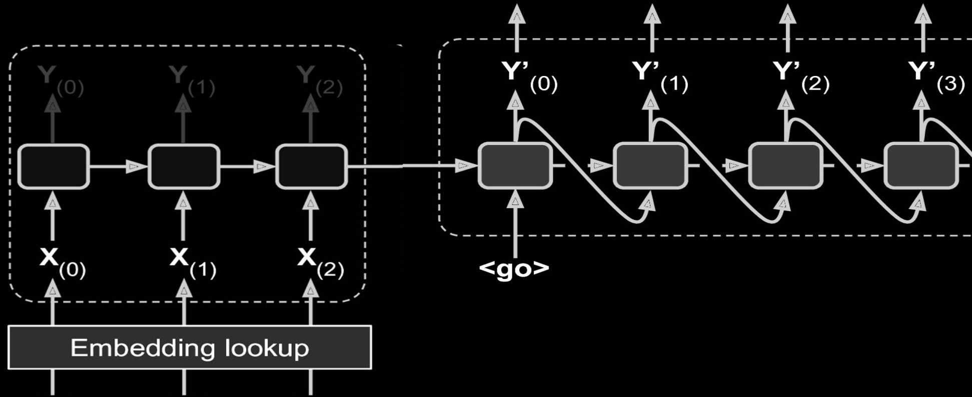


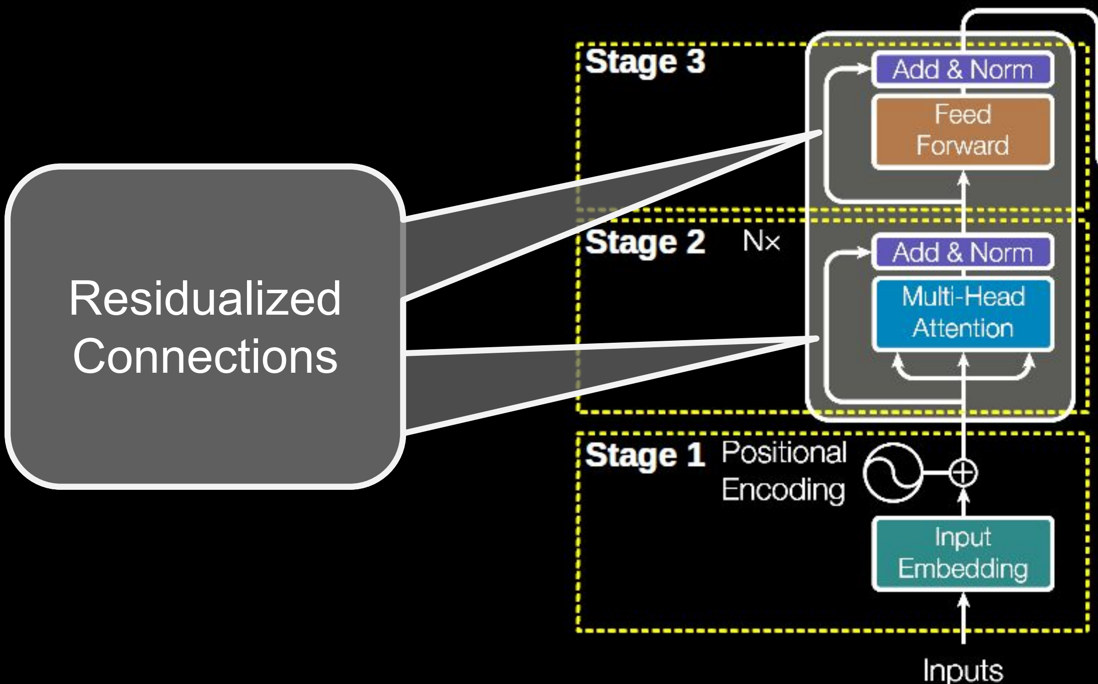
Transformer for Encoder-Decoder



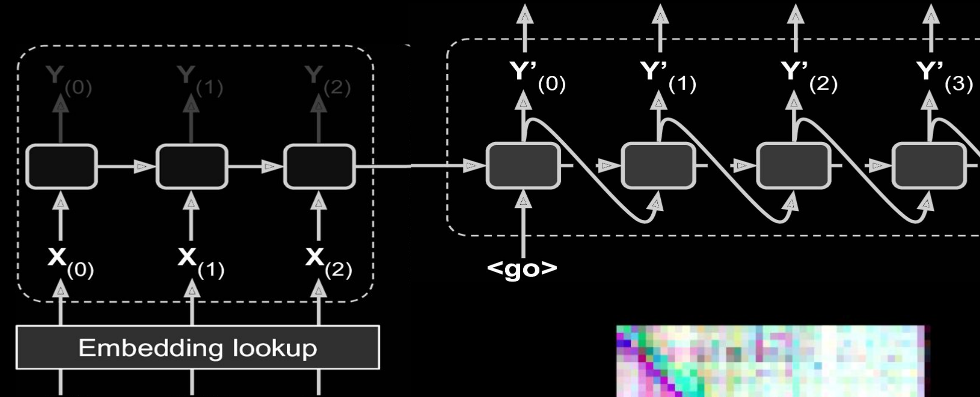


Transformer for Encoder-Decoder

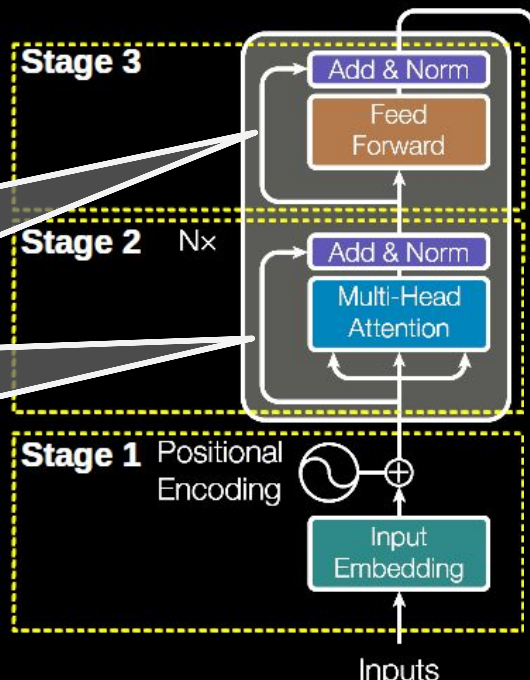




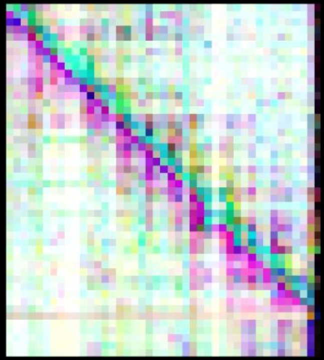
Transformer for Encoder-Decoder



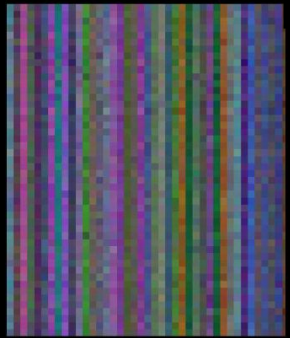
Residualized Connections



residuals enable positional information to be passed along

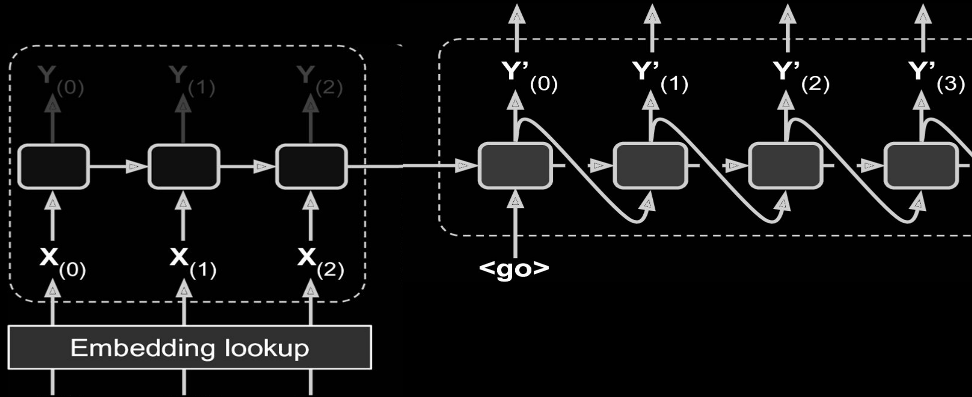


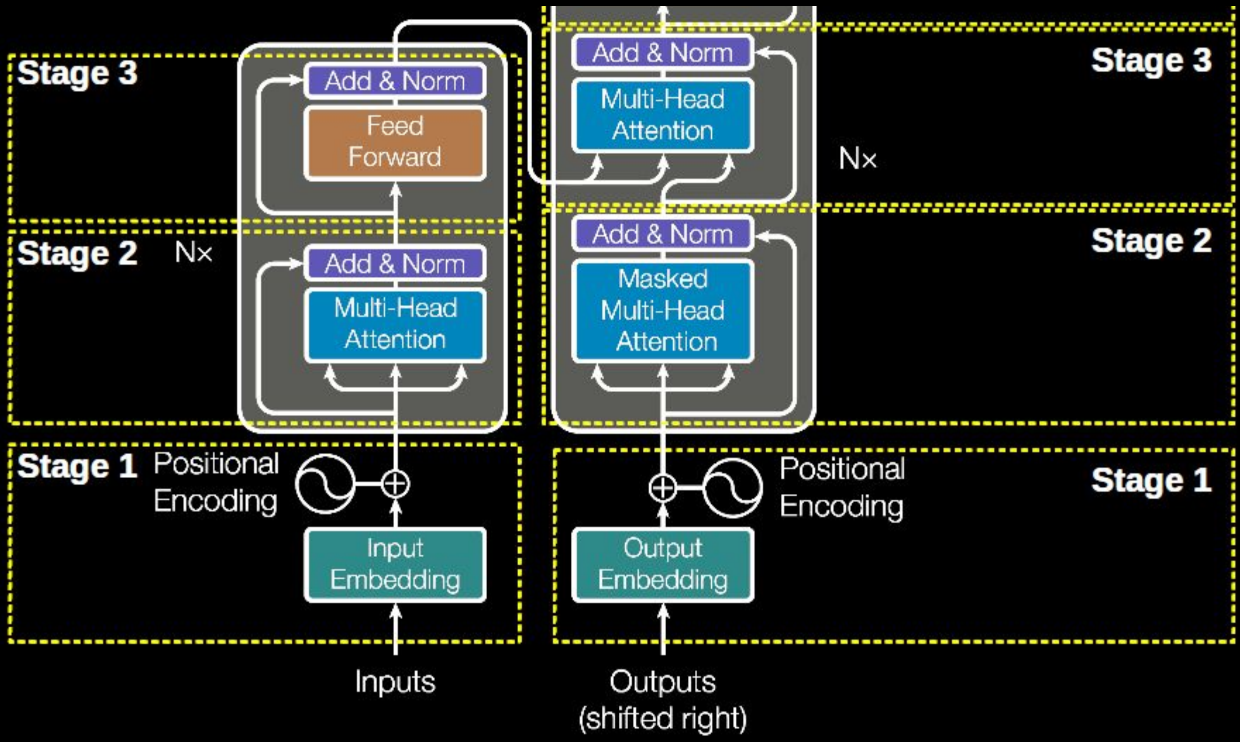
With residuals



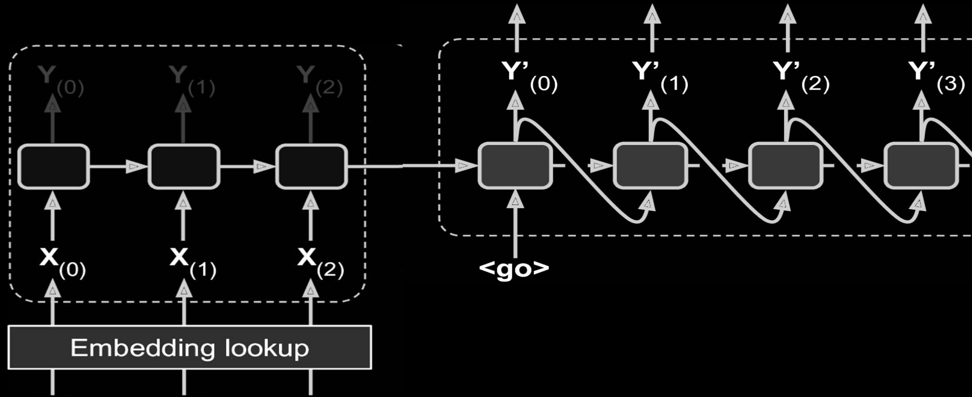
Without residuals

Transformer for Encoder-Decoder

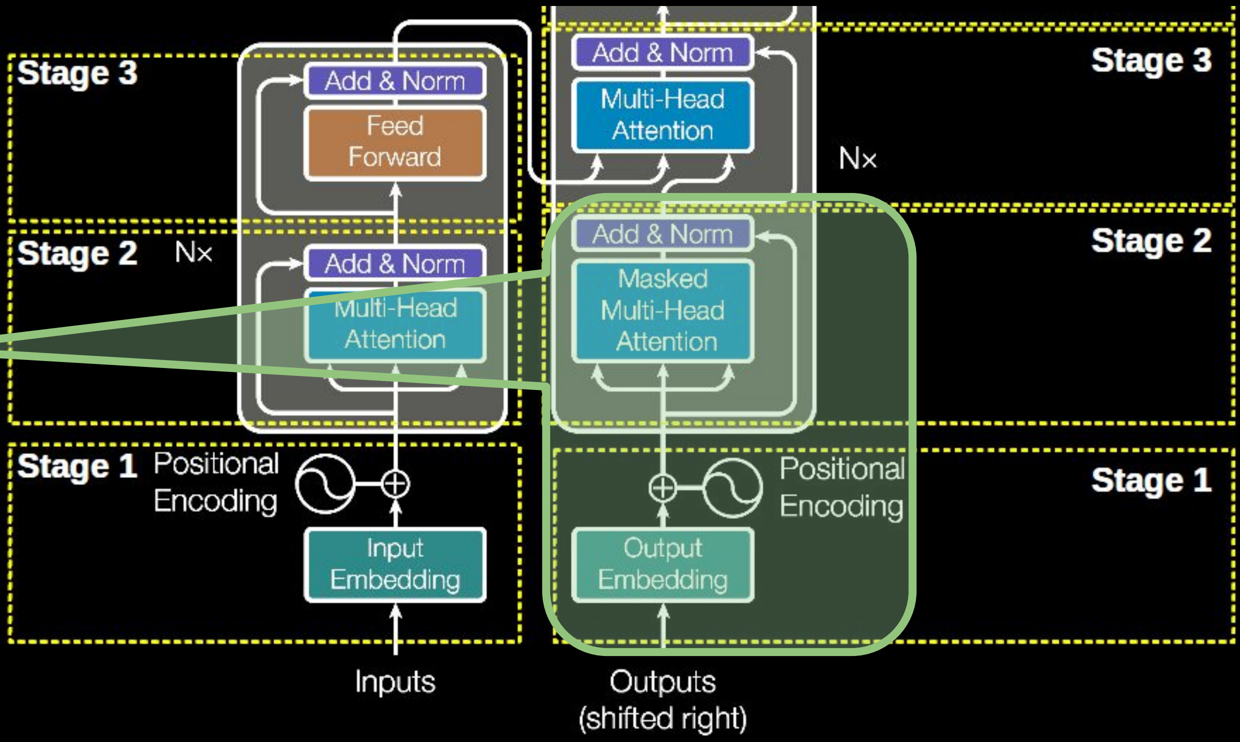




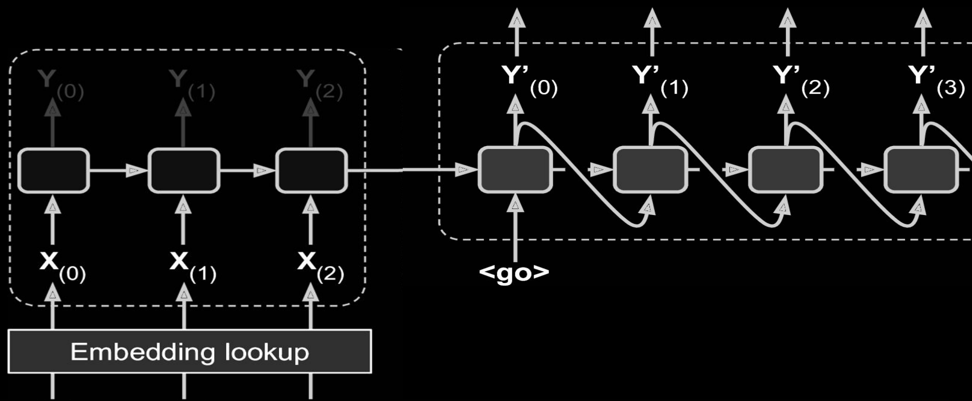
Transformer for Encoder-Decoder



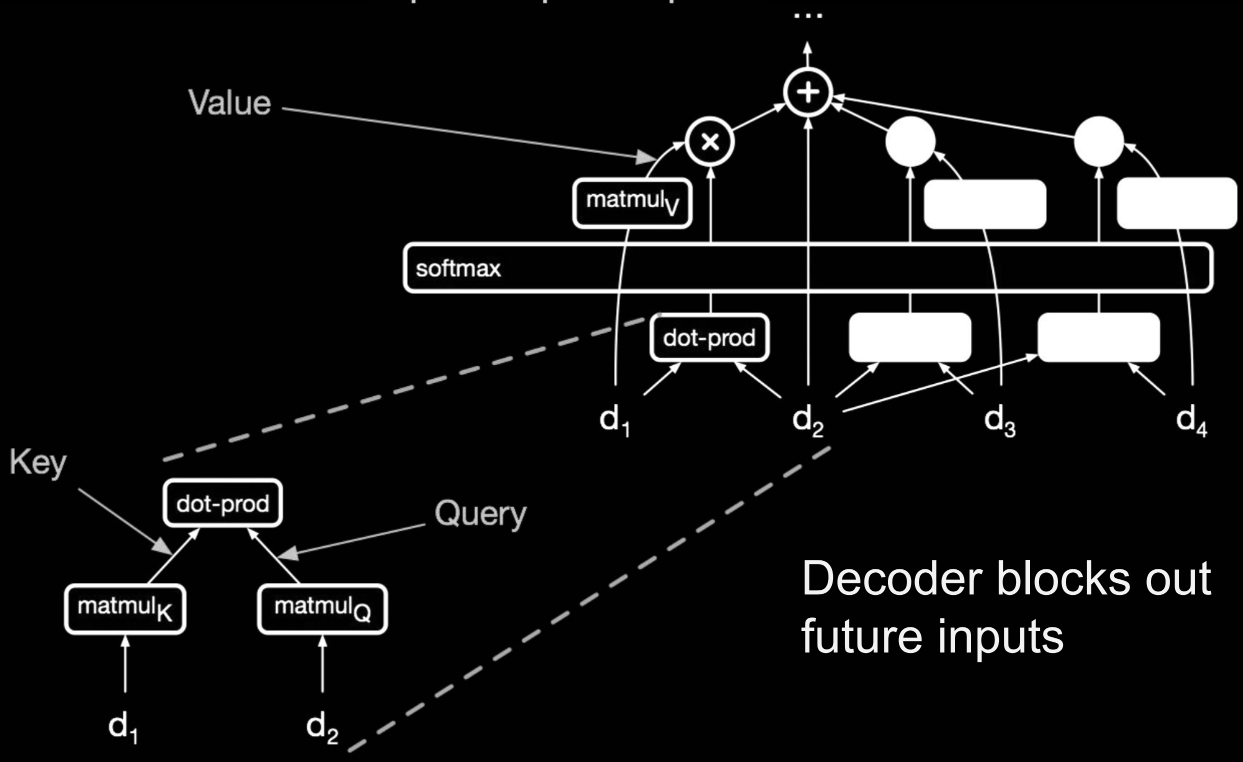
essentially, a language model



Transformer for Encoder-Decoder



essentially, a language model

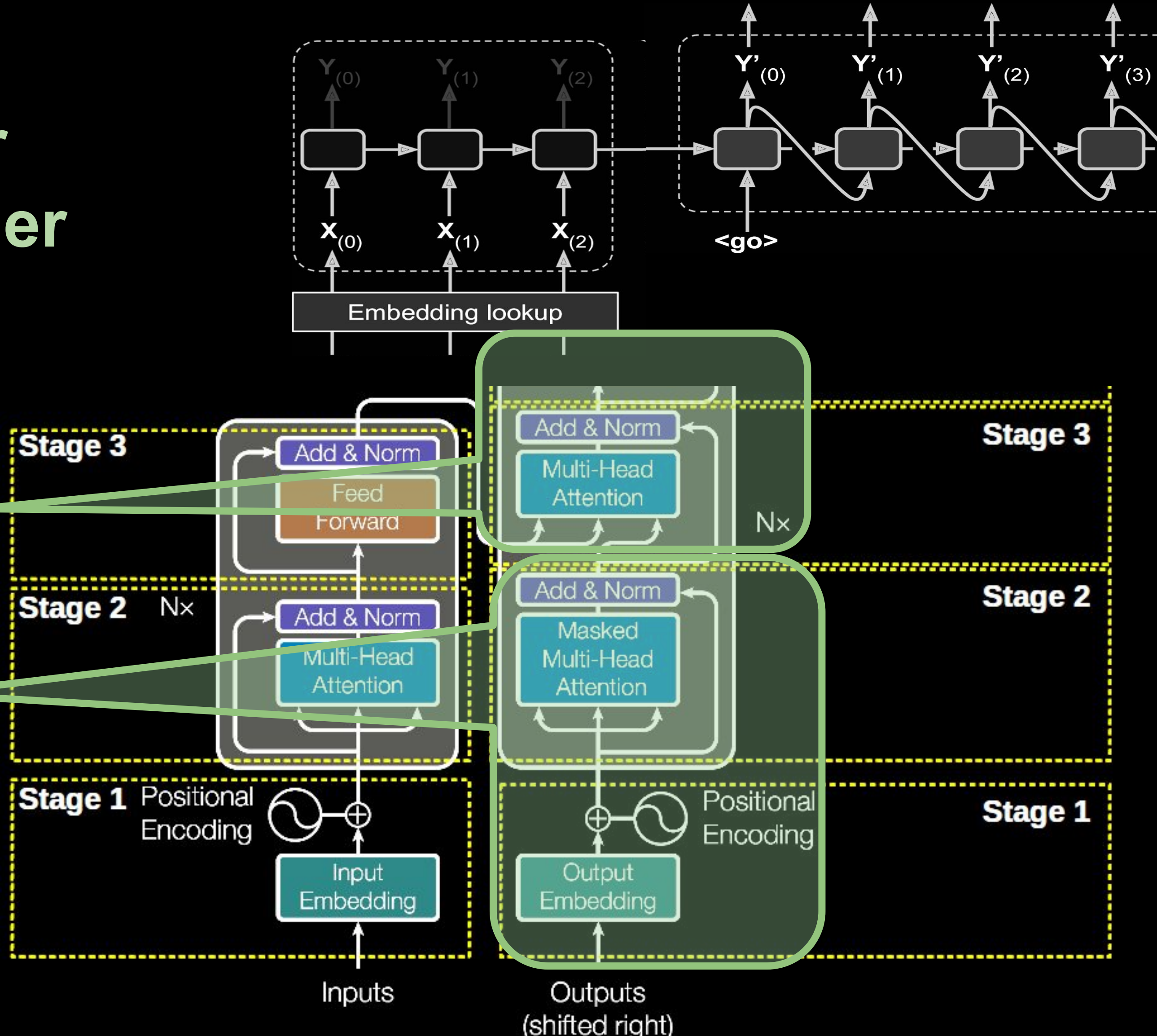


Decoder blocks out future inputs

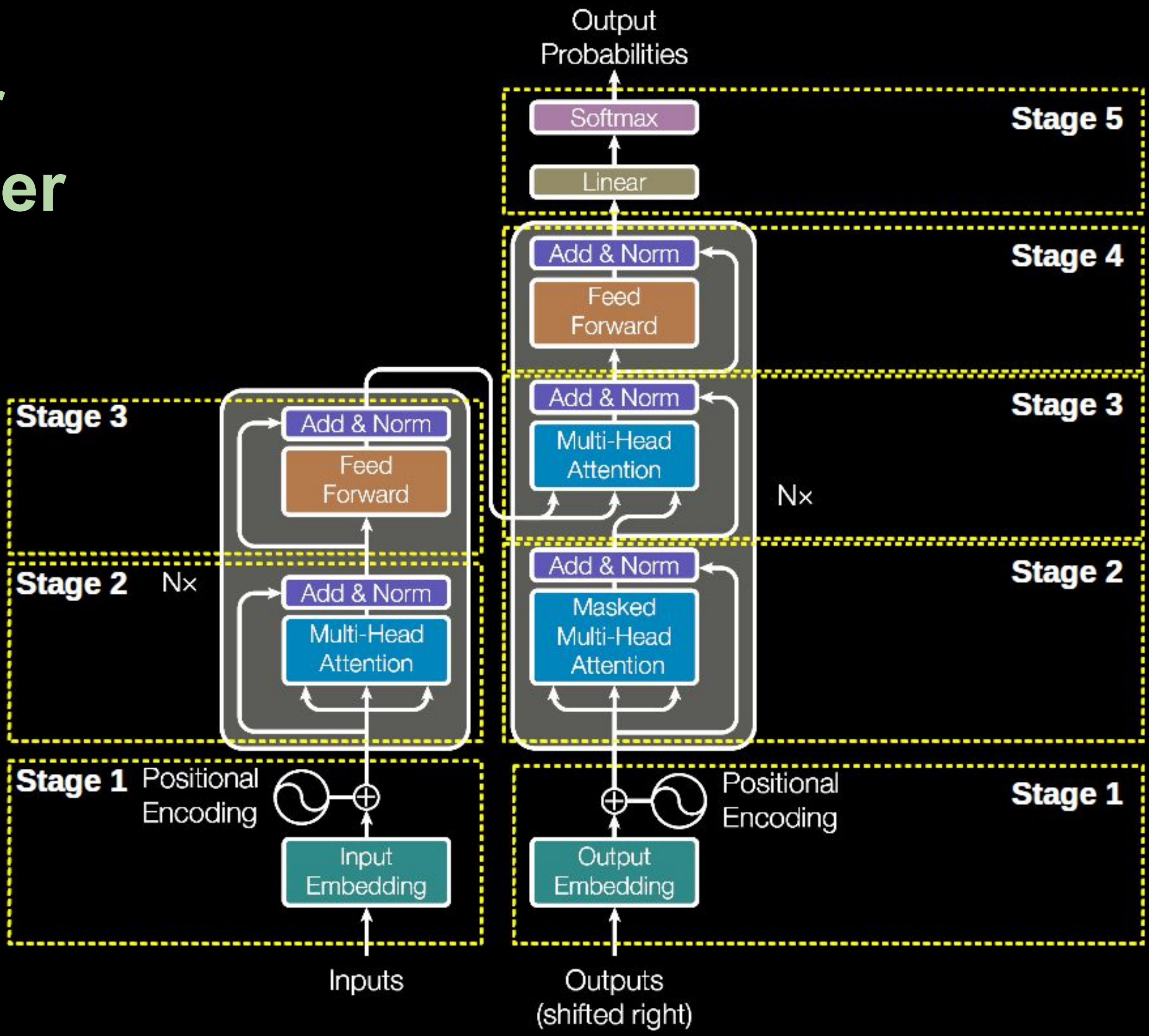
Transformer for Encoder-Decoder

Add conditioning of the LM based on the encoder

essentially, a language model



Transformer for Encoder-Decoder



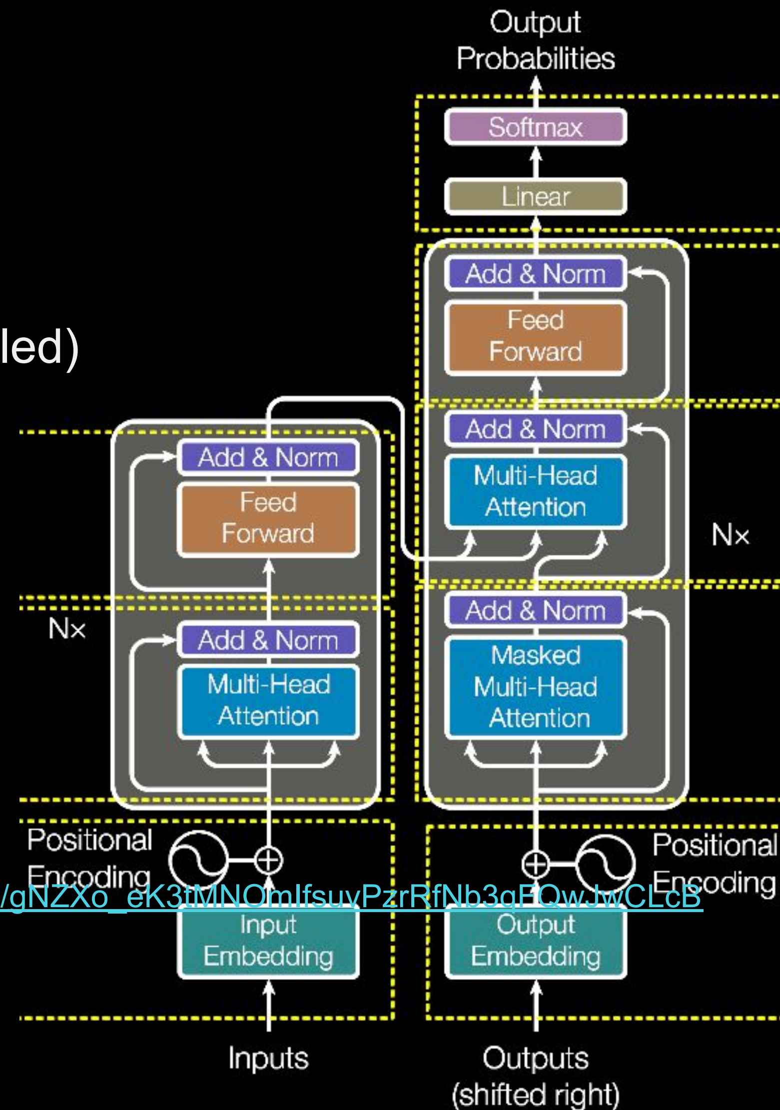
Transformer (as of 2017)

“WMT-2014” Data Set. BLEU scores:

	EN-DE	EN-FR
GNMT (orig)	24.6	39.9
ConvSeq2Seq	25.2	40.5
Transformer*	**28.4**	**41.8**

Transformer

- Utilize Self-Attention
- Simple att scoring function (dot product, scaled)
- Added linear layers for Q, K, and V
- Multi-head attention
- Added positional encoding
- Added residual connection
- Simulate decoding by masking



https://4.bp.blogspot.com/-OlrV-PAtEkQ/W3RkOJCBkaI/AAAAAAAADOg/gNZXo_eK3tMNOmIfsuvPzrRfNb3qFQwJwCLcB GAs/s640/image1.gif

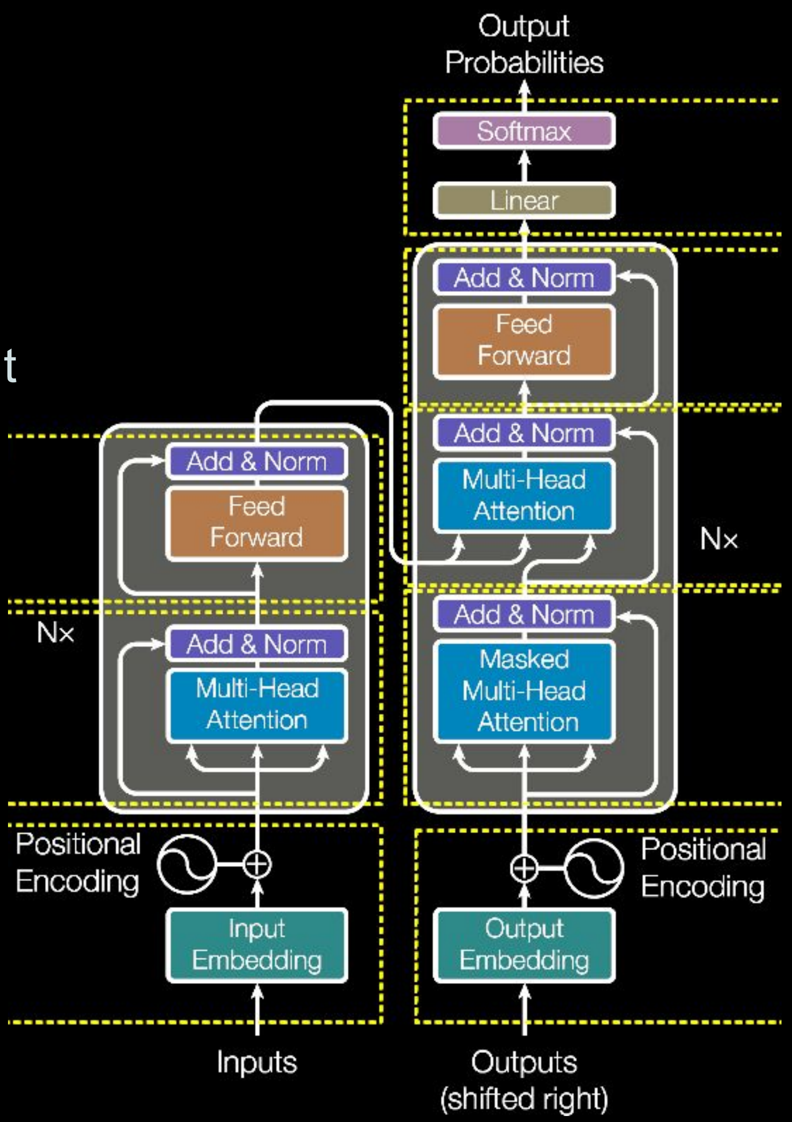
Transformer

Why?

- Don't need complexity of LSTM/GRU cells
- Constant num edges between words (or input steps)
- Enables "interactions" (i.e. adaptations) between words
- Easy to parallelize -- don't need sequential processing.

Drawbacks:

- Only unidirectional by default
- Only a "single-hop" relationship per layer (multiple layers to capture multiple)



BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

Produces contextualized embeddings
(or pre-trained contextualized encoder)

Drawbacks of Vanilla Transformers:

- Only unidirectional by default
- Only a "single-hop" relationship per layer
  (multiple layers to capture multiple)

BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

Produces contextualized embeddings
(or pre-trained contextualized encoder)

- Bidirectional context by “masking” in the middle
- A lot of layers, hidden states, attention heads.

Drawbacks of Vanilla Transformers:

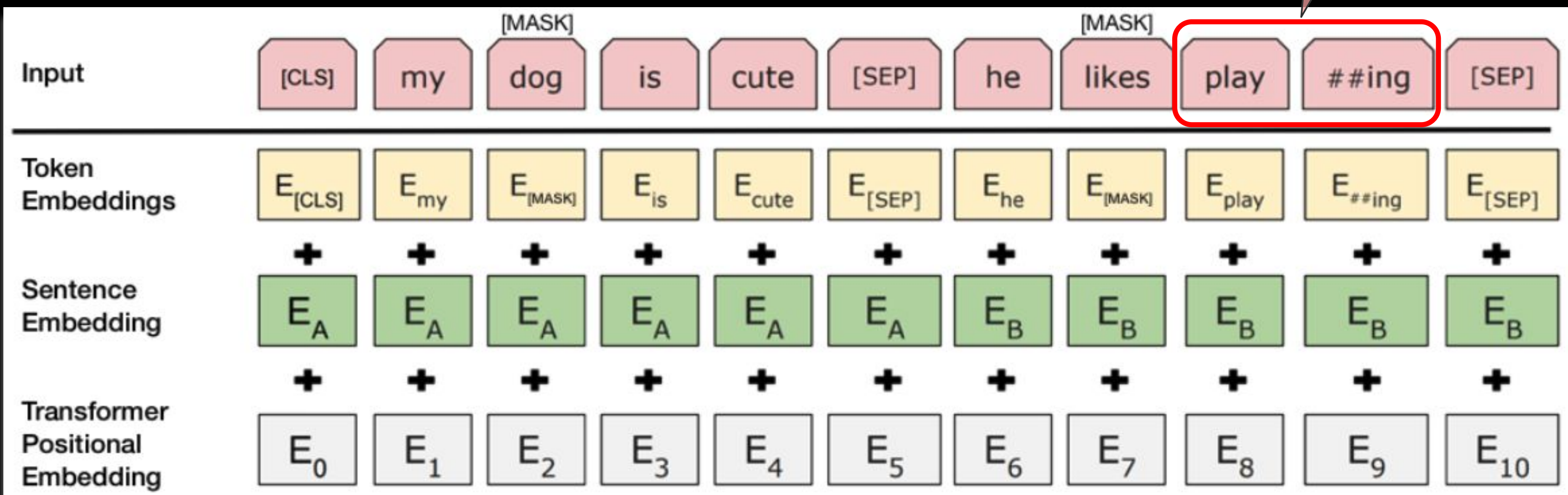
- Only unidirectional by default
- Only a “single-hop” relationship per layer
  (multiple layers to capture multiple)

BERT

tokenize into “word pieces”

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

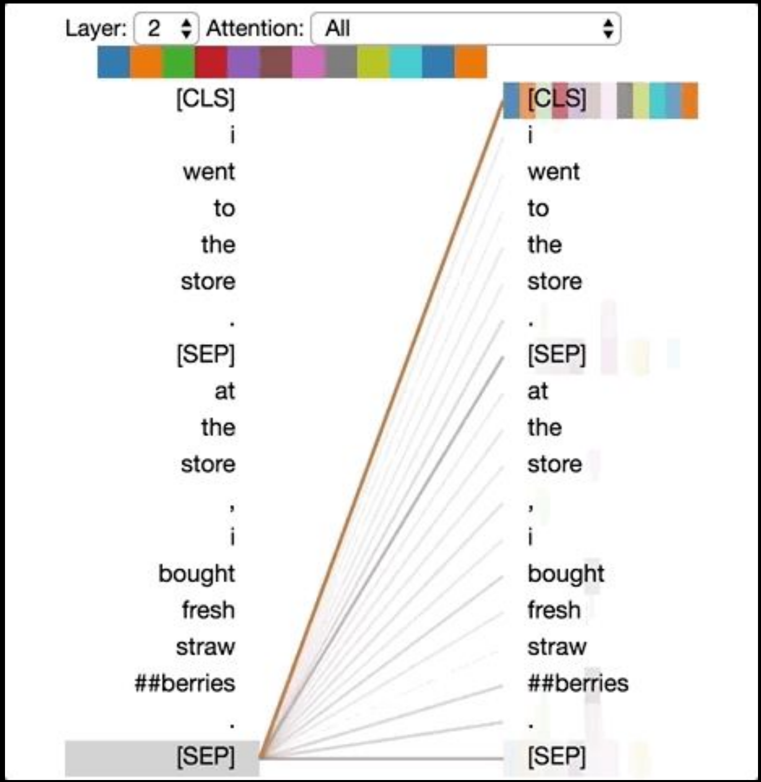
Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`



(Devlin et al., 2019)

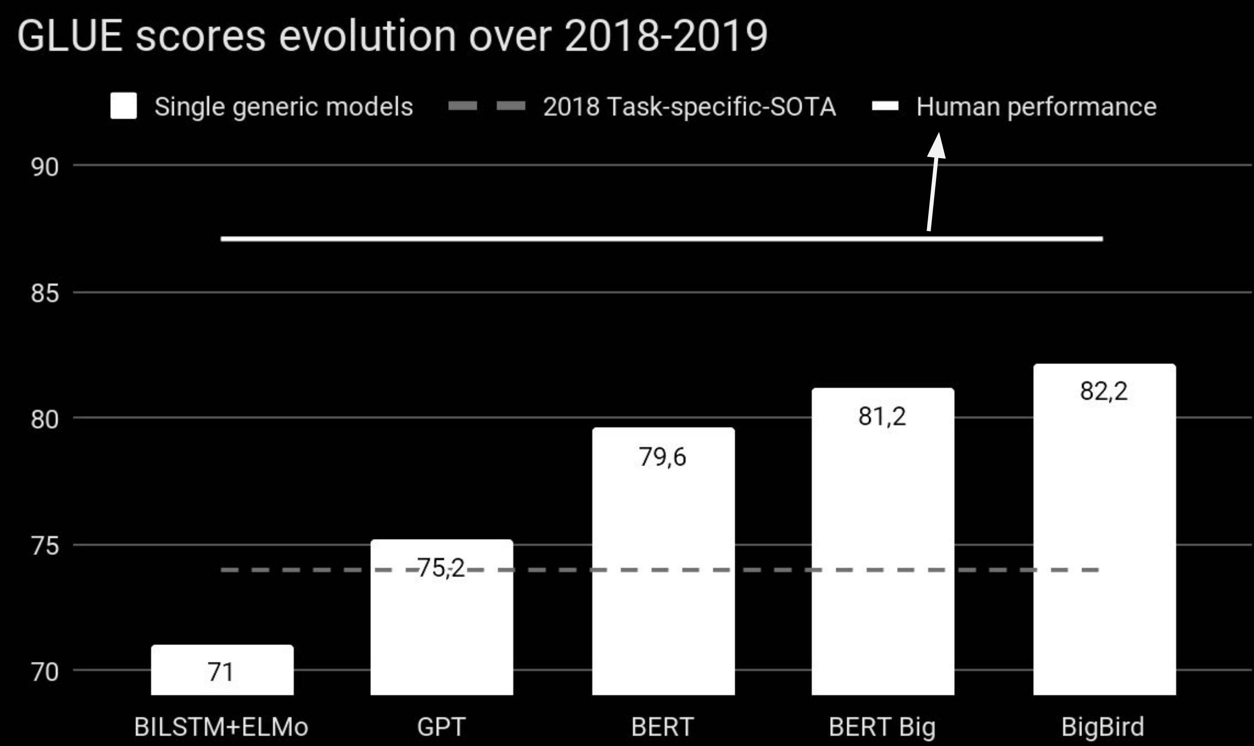
Bert: Attention by Layers

https://colab.research.google.com/drive/1vlOJ1lhdujVjfH857hvYKIdKPTD9Kid8



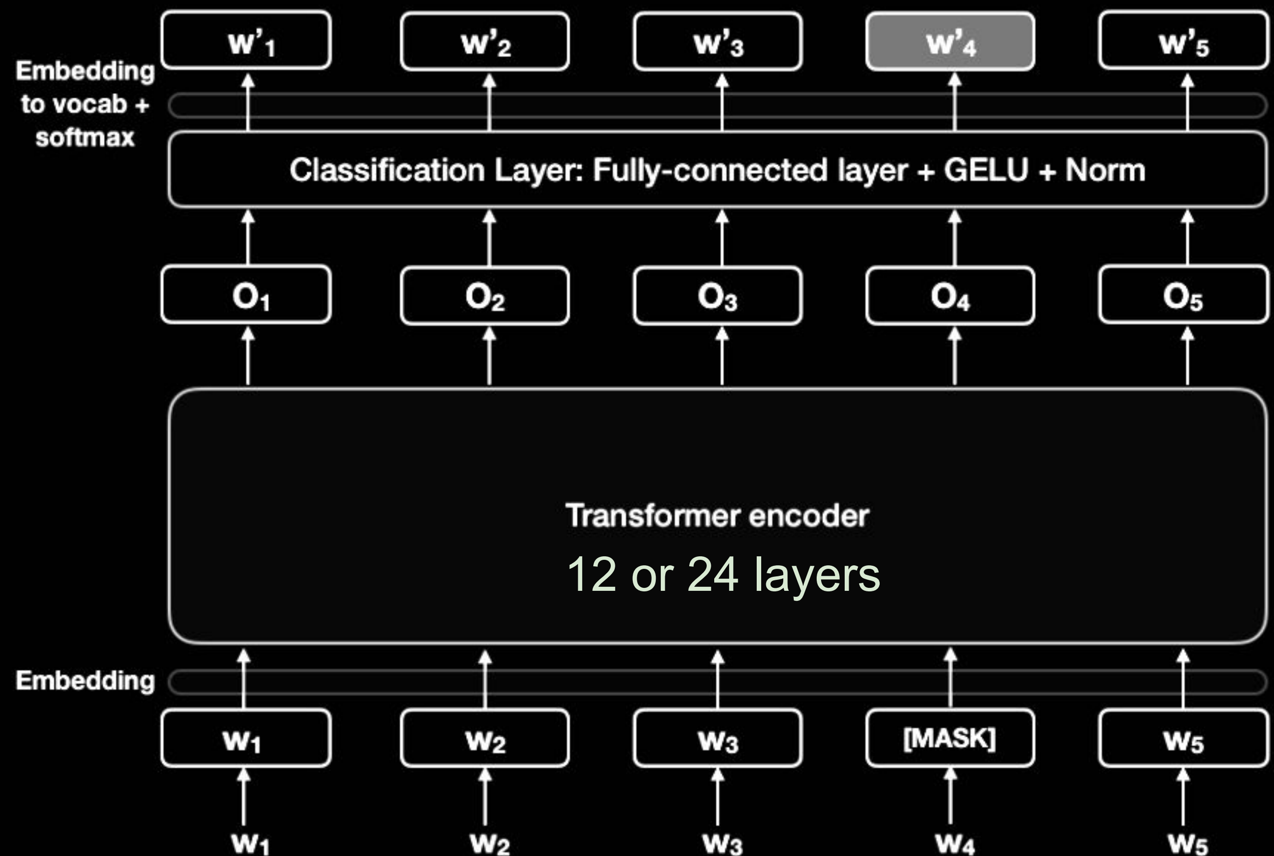
(Vig, 2019)

BERT Performance: e.g. Question Answering

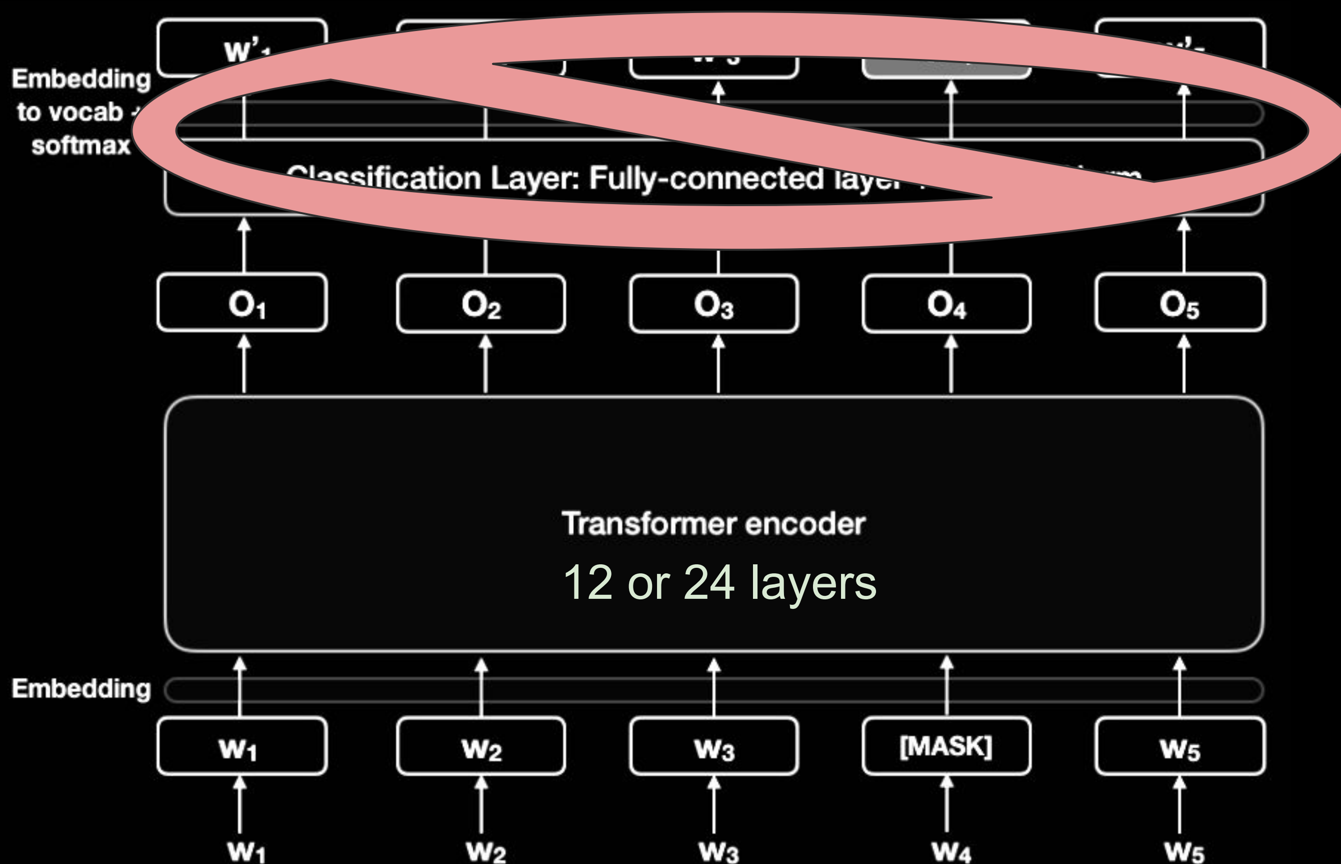


https://rajpurkar.github.io/SQuAD-explorer/

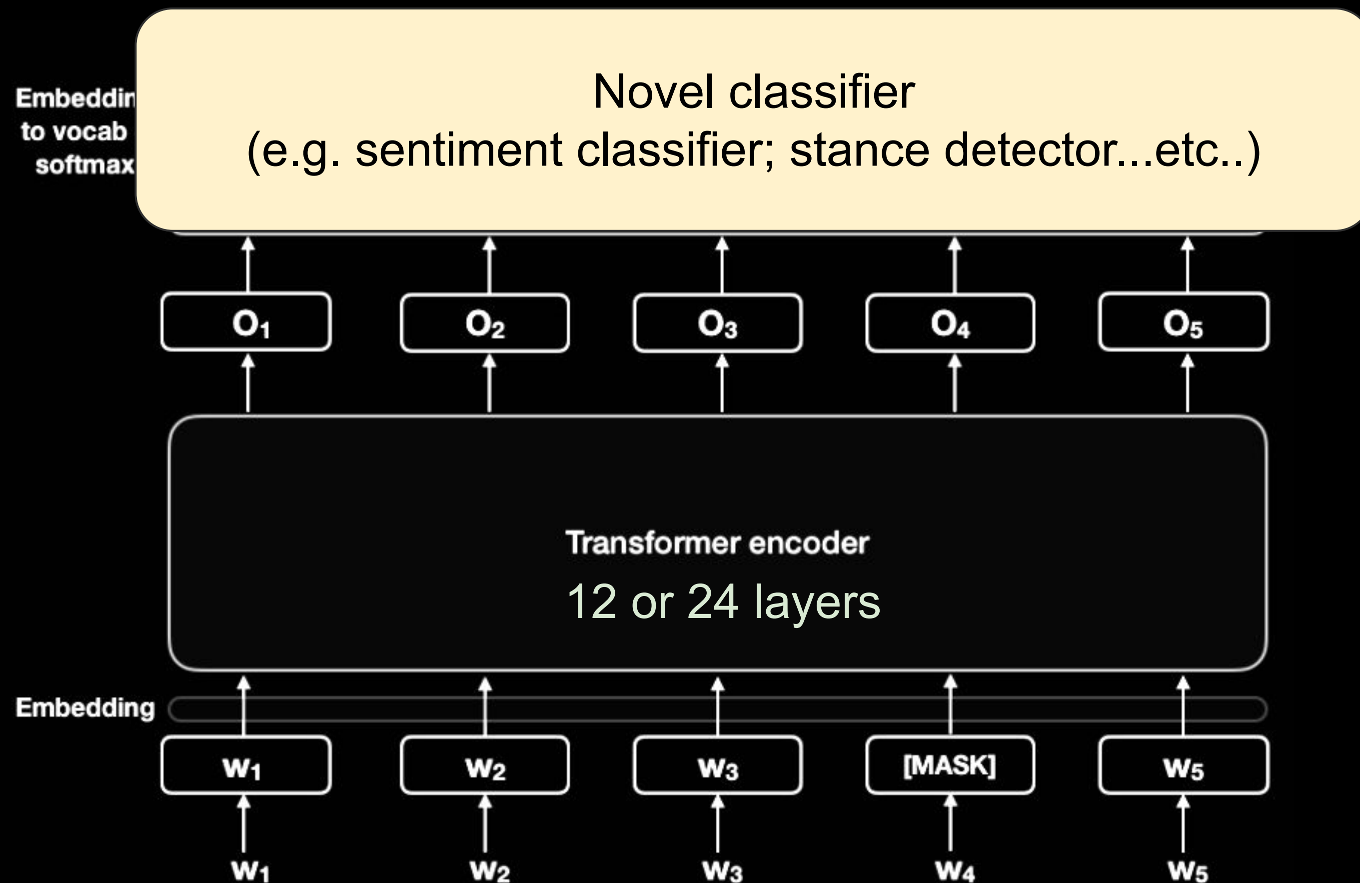
BERT: Pre-training; Fine-tuning



BERT: Pre-training; Fine-tuning



BERT: Pre-training; Fine-tuning



The Transformer: "Attention-only" models

Can handle sequences and long-distance dependencies, but….

- Don't want complexity of LSTM/GRU cells
- Constant num edges between input steps
- Enables "interactions" (i.e. adaptations) between words
- **Easy to parallelize -- don't need sequential processing.**