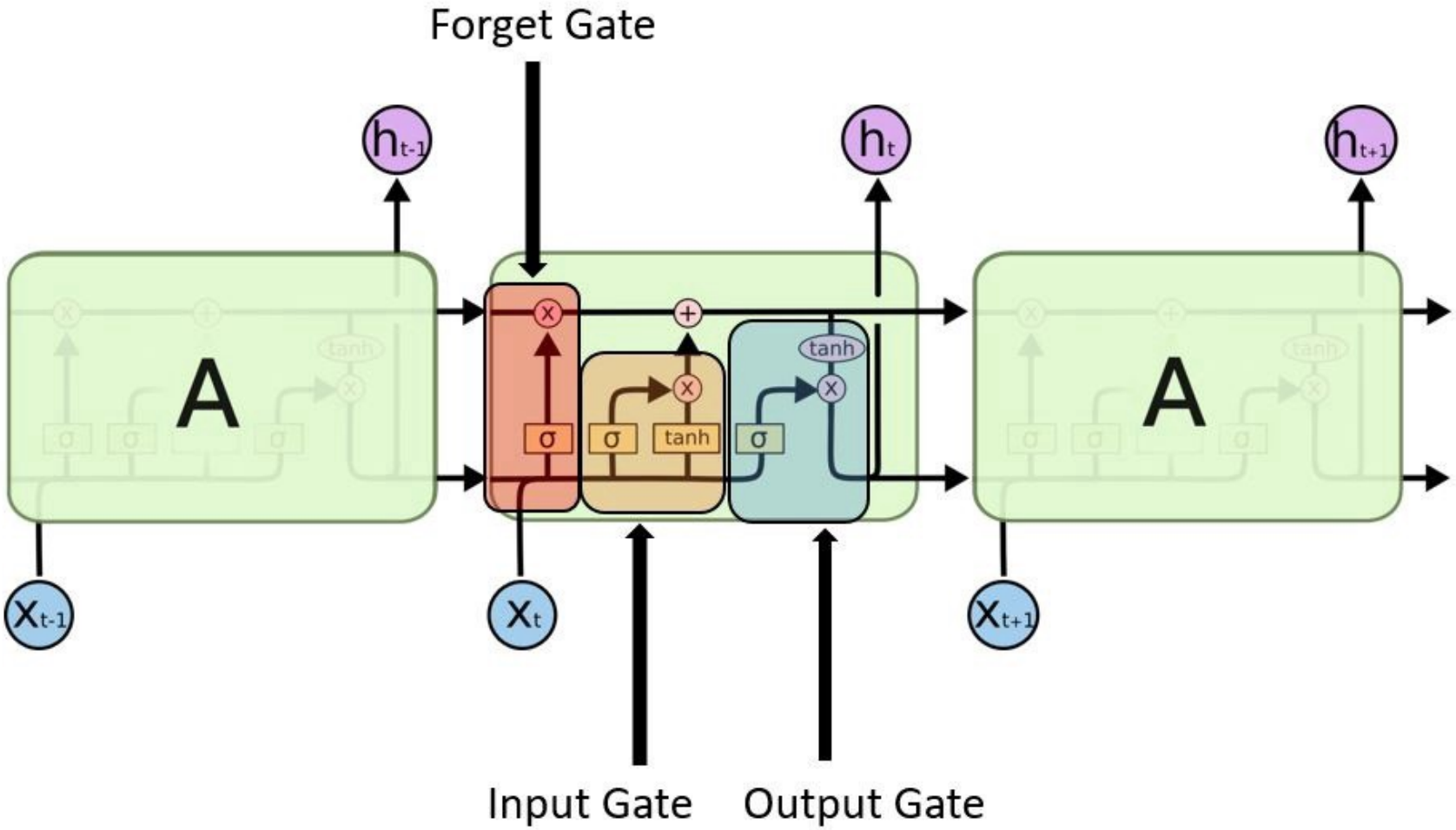
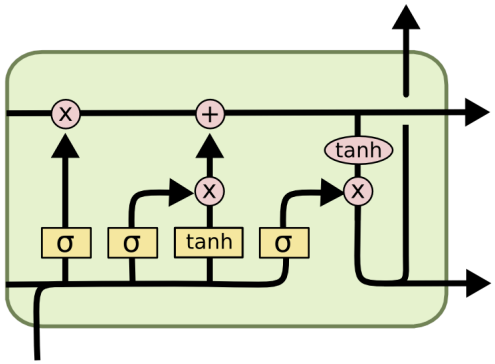
Deep Learning

Mayank Vatsa


Forget Gate
h_{t-1}
h_t
h_{t+1}
A
A
tanh
σ
σ
tanh
σ
tanh
X_{t-1}
X_t
X_{t+1}
Input Gate
Output Gate



These 4 matrix computation should be done concurrently.

$z = tanh(W \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$

c^{t-1}

$z^i = \sigma(W^i \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$

Controls forget gate

Controls input gate

Updating information

Controls Output gate

$z^f = \sigma(W^f \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$

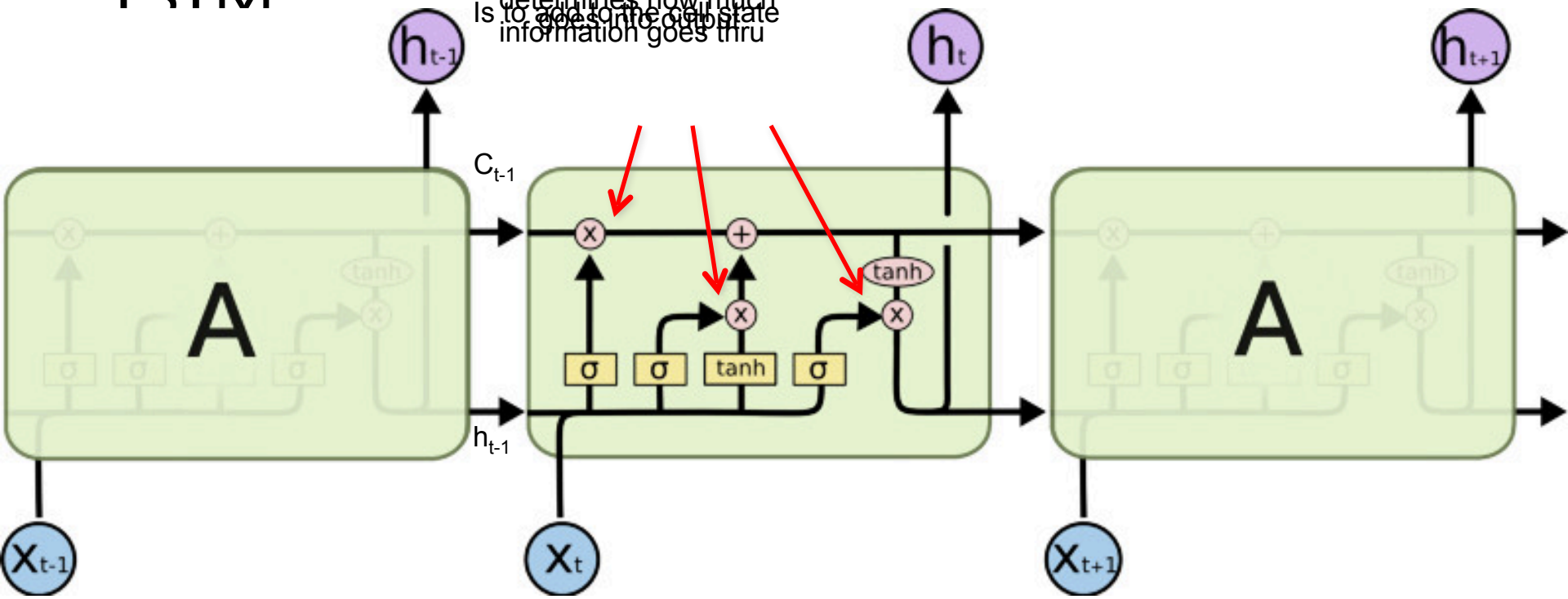
z^f z^i z z^o

h^{t-1} x^t

$z^o = \sigma(W^o \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$

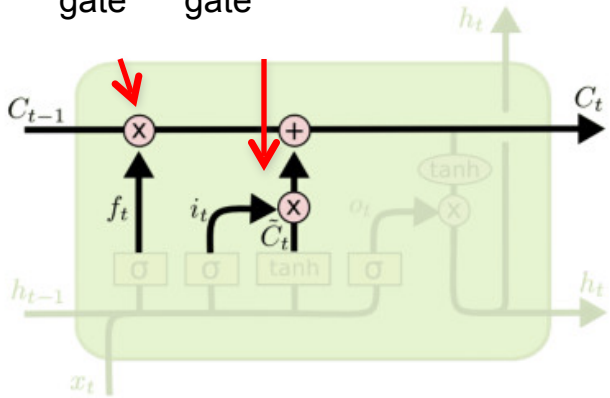
Information flow of LSTM

LSTM



Forget gate

input gate

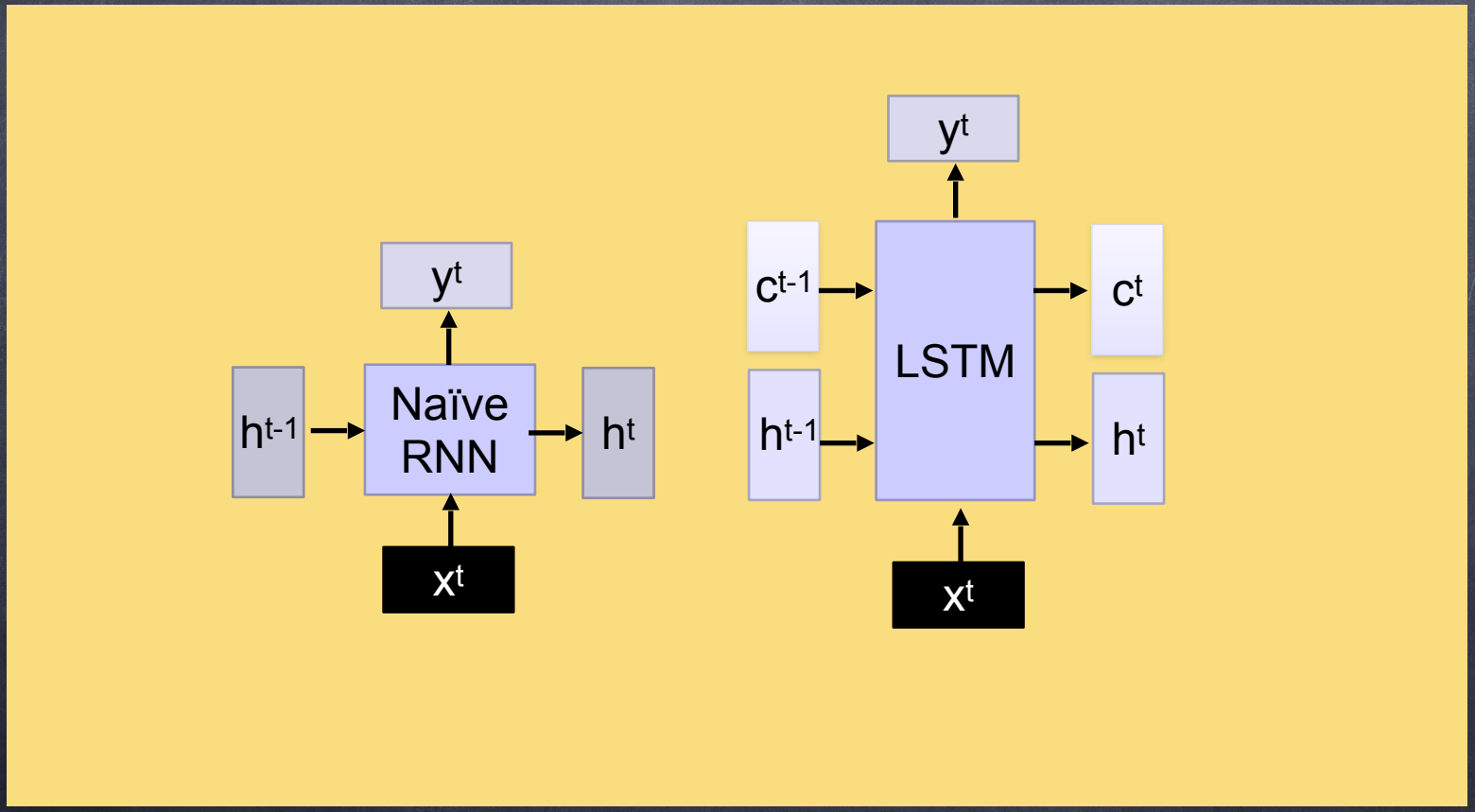


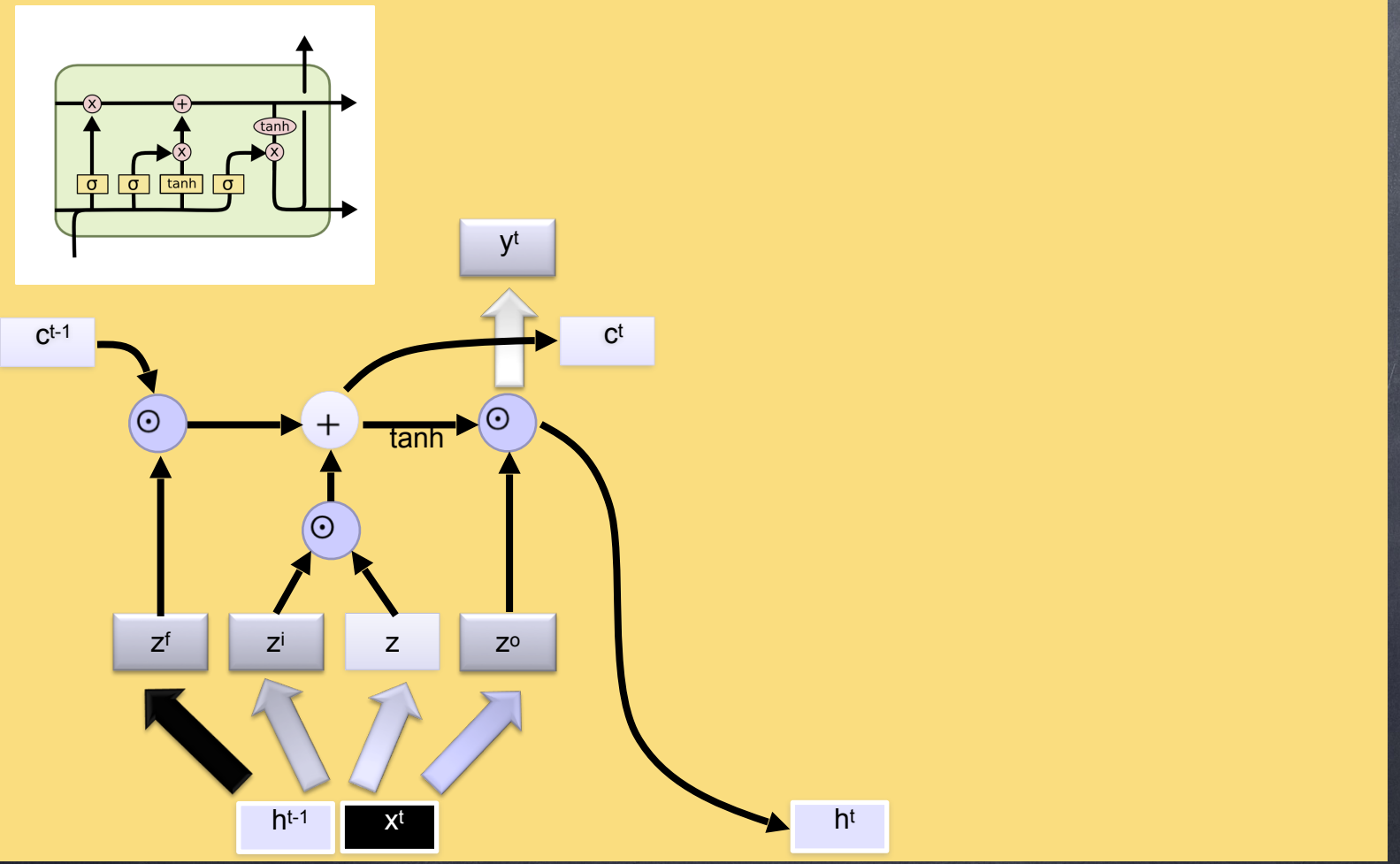
The core idea is this cell state C_t, it is changed slowly, with only minor linear interactions. It is very easy for information to flow along it unchanged.

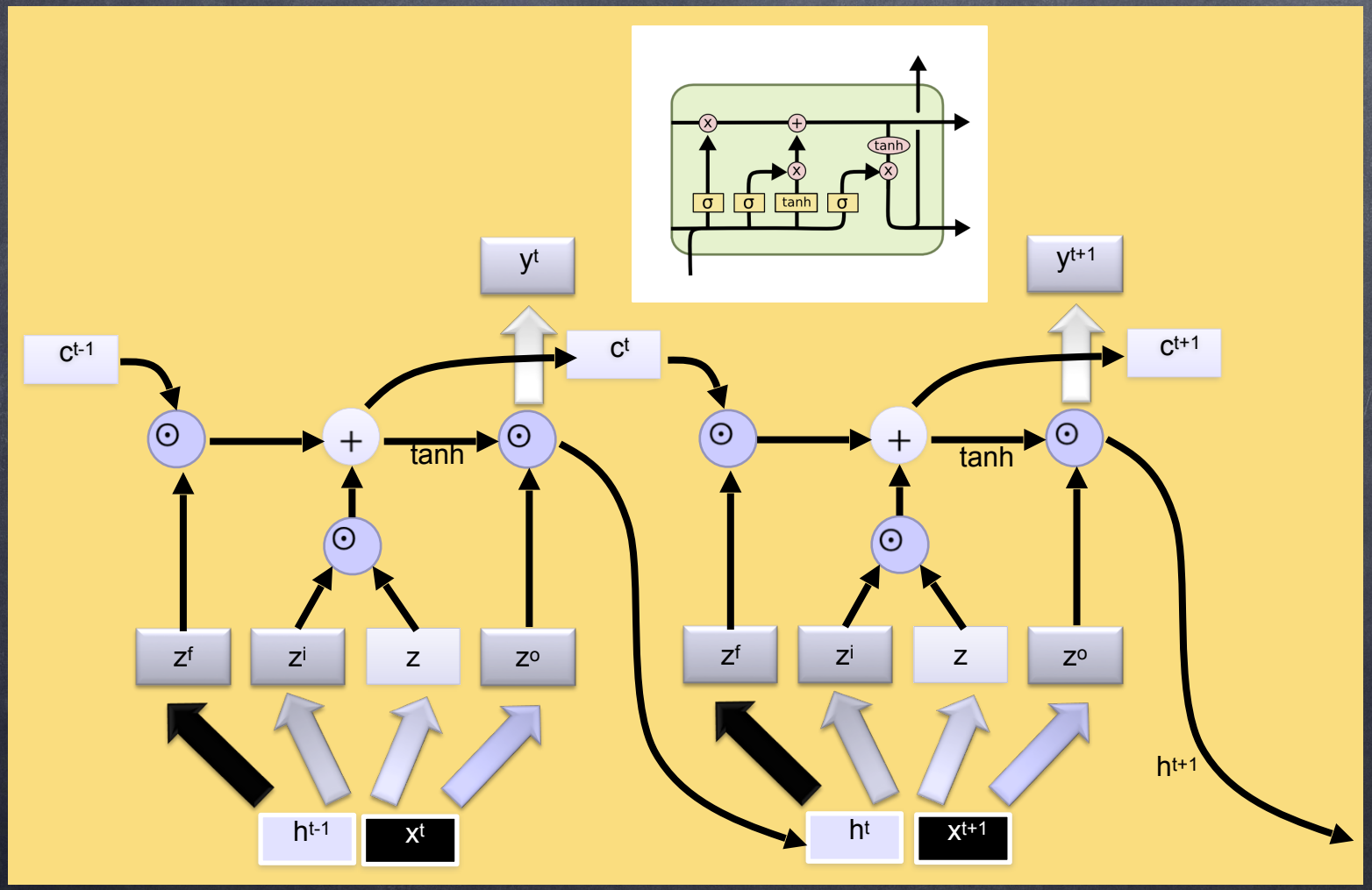
Why sigmoid or tanh:
Sigmoid: 0,1 gating as switch.
Vanishing gradient problem in LSTM is handled already.
ReLU replaces tanh

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

RNN vs LSTM




σ
σ
tanh
σ
tanh
y^t
c^{t-1}
c^t
tanh
z^f
z^i
z
z^o
h^{t-1}
x^t
h^t


σ
σ
tanh
σ
tanh
y^t
y^{t+1}
c^{t-1}
c^t
c^{t+1}
tanh
tanh
z^f
z^i
z
z^o
z^f
z^i
z
z^o
h^{t-1}
x^t
h^t
x^{t+1}
h^{t+1}

Applications of LSTM / RNN

a train traveling down a track next to a forest.

a group of young boys playing soccer on a field.

Image caption generation using attention (From CY Lee lecture)

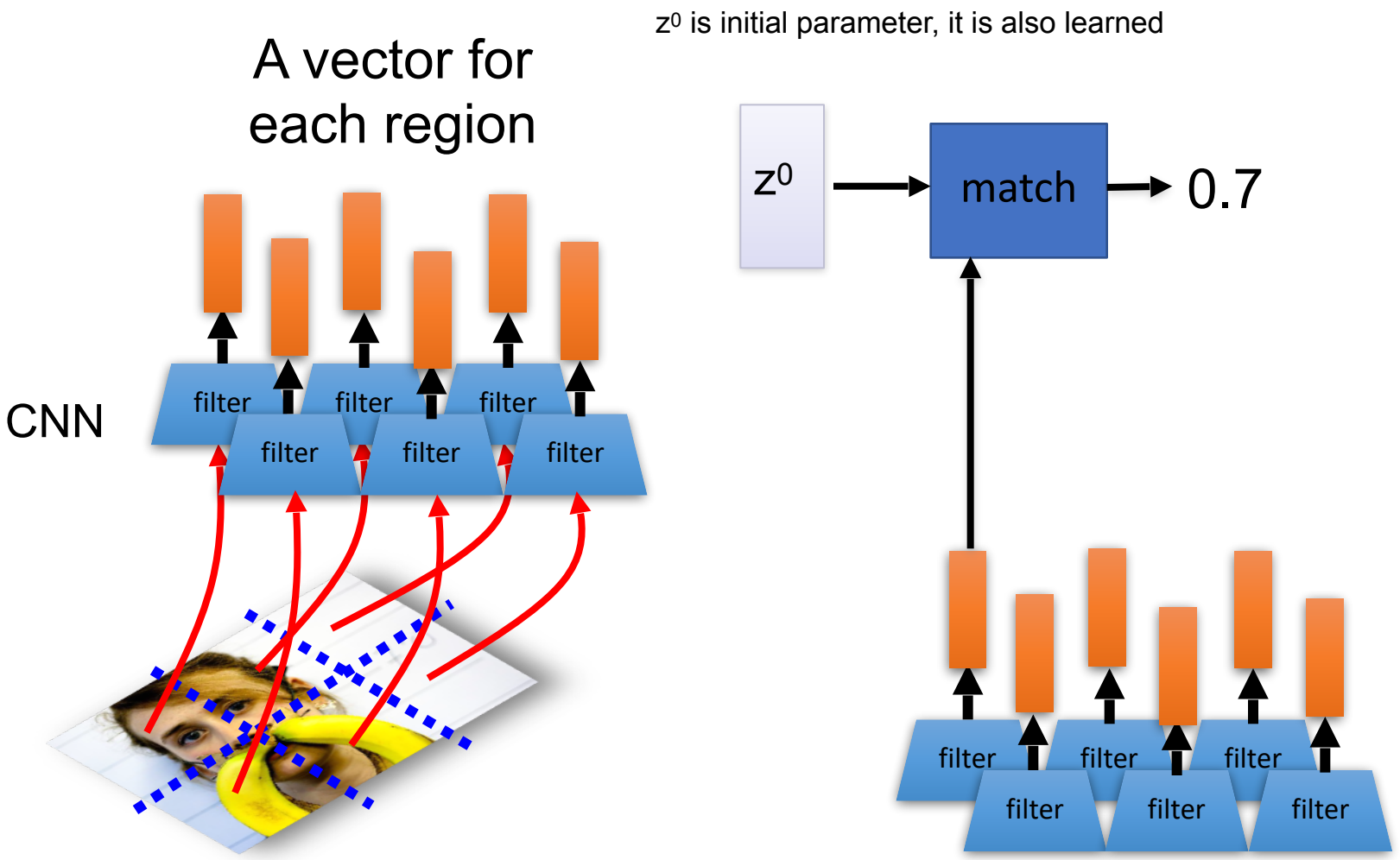


Image Caption Generation

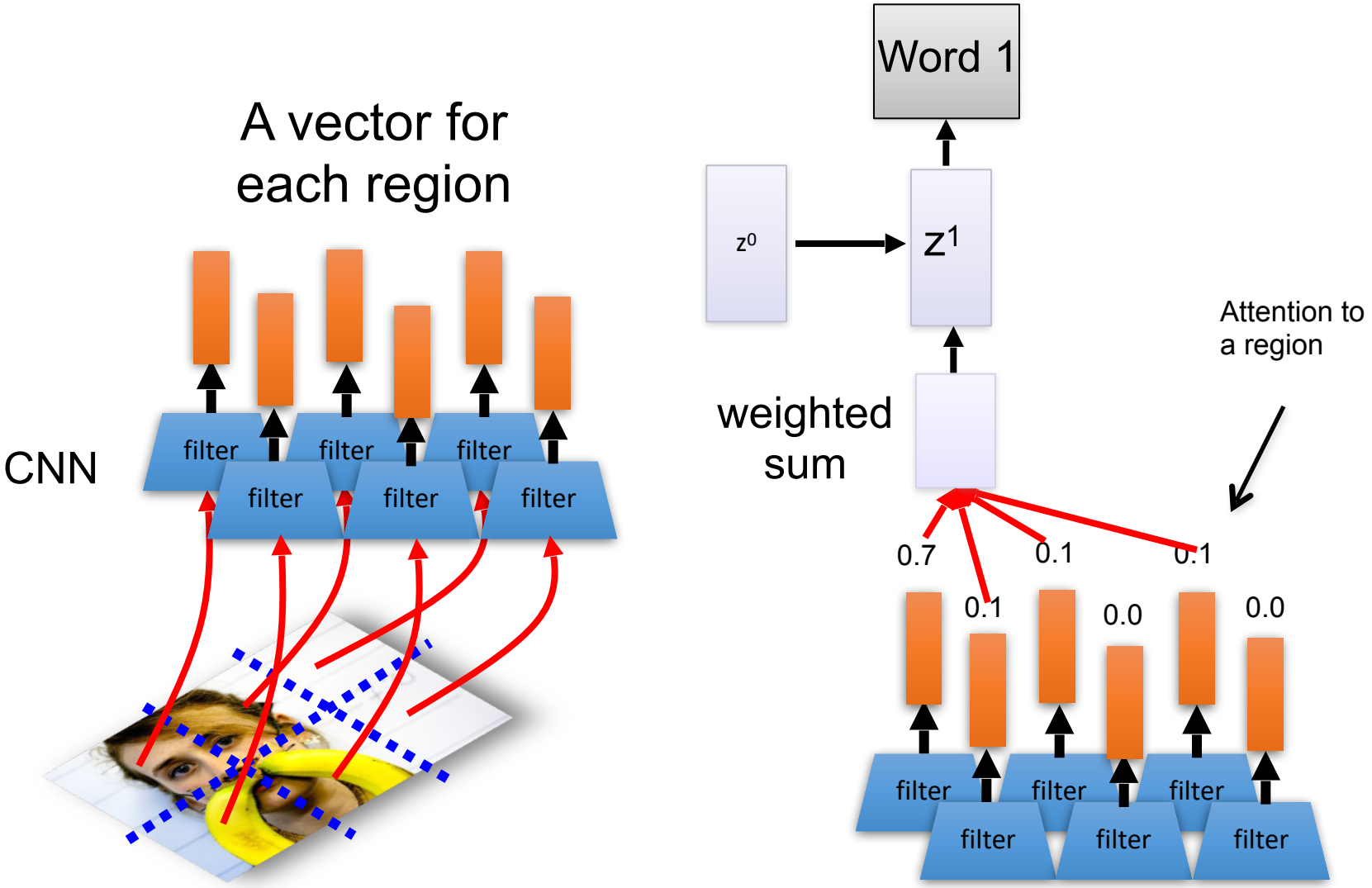


Image Caption Generation

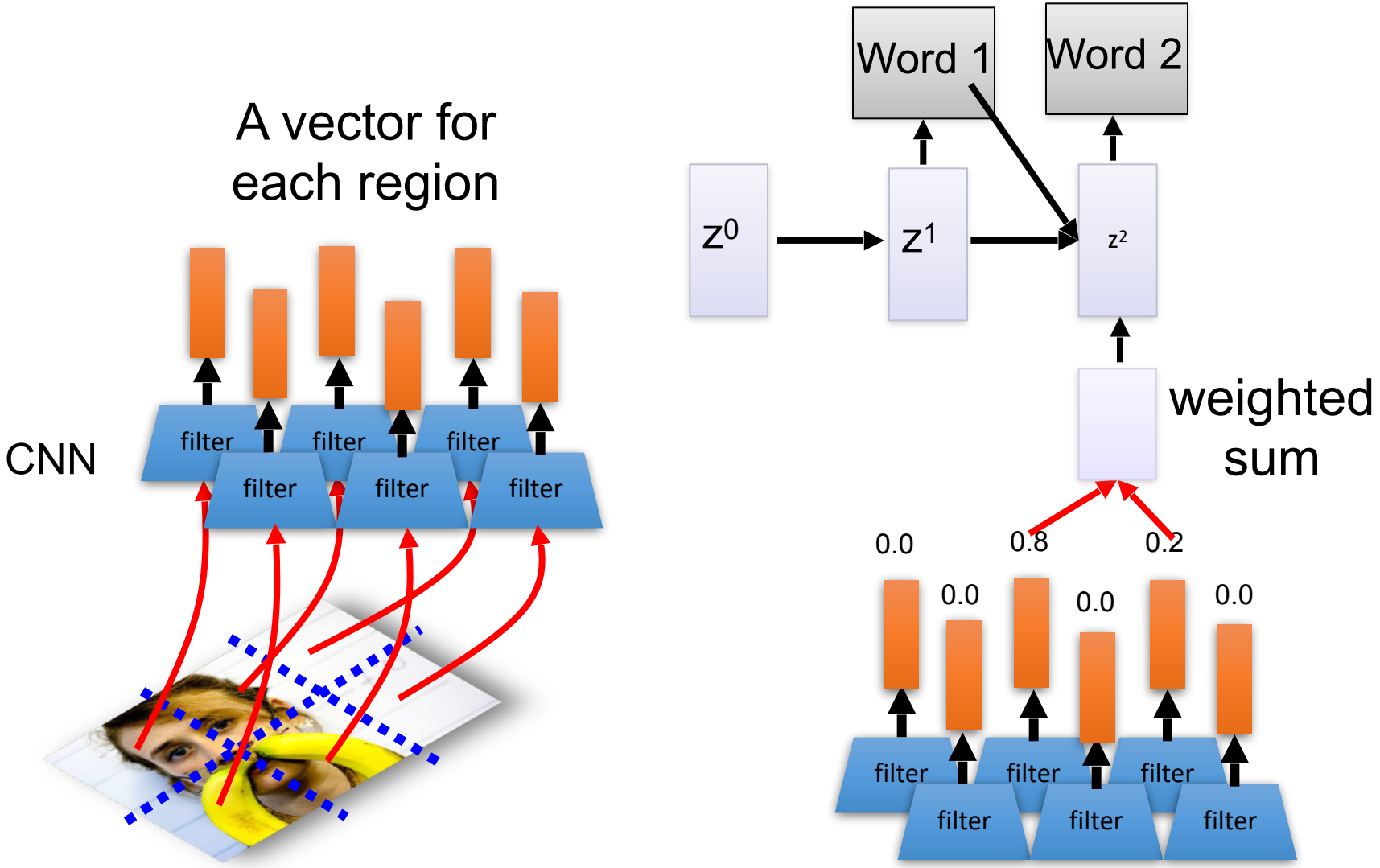


Image Caption Generation

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML, 2015

Image Caption Generation

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

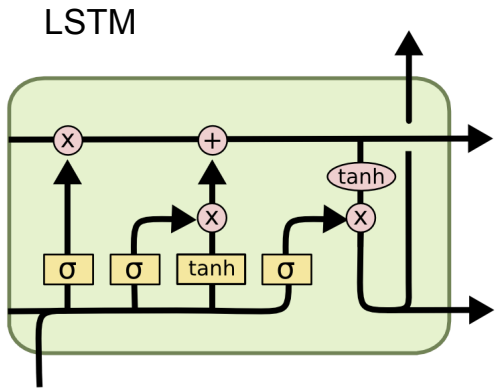
A woman is sitting at a table
with a large pizza.

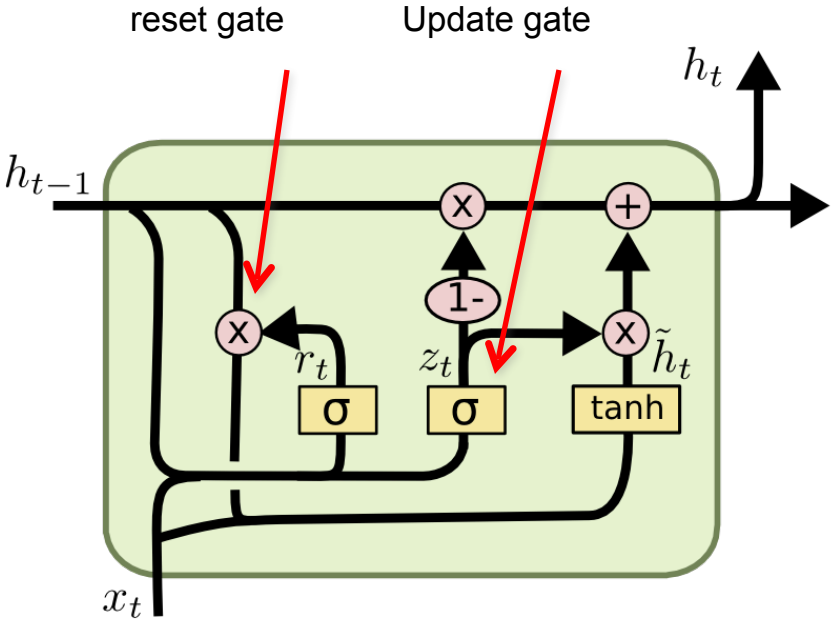
A man is talking on his cell phone
while another man watches.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”, ICML, 2015

GRU – gated recurrent unit

(more compression)





$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

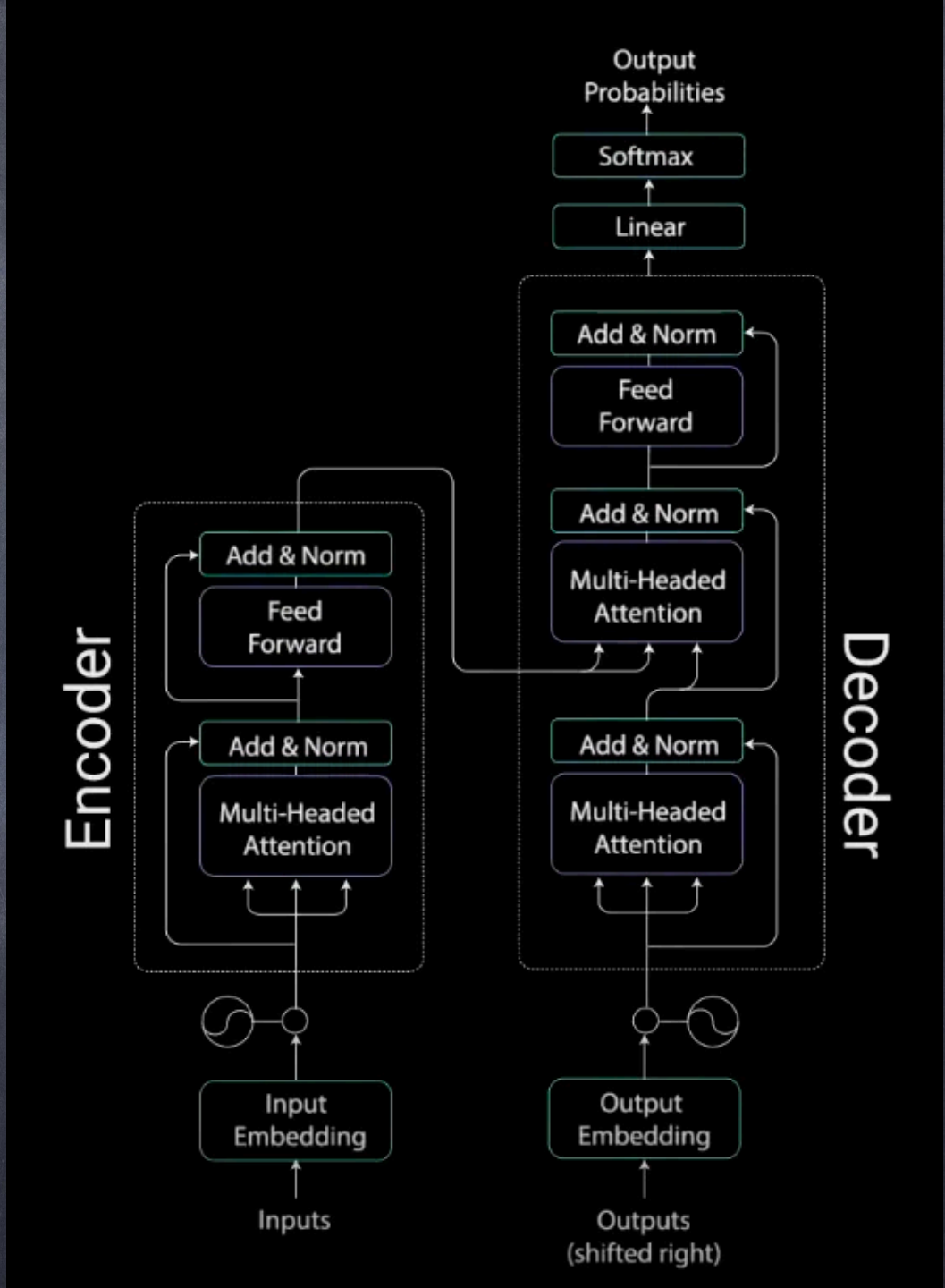
It combines the forget and input into a single update gate.
It also merges the cell state and hidden state. This is simpler than LSTM. There are many other variants too.

X,*: element-wise multiply

More advancements

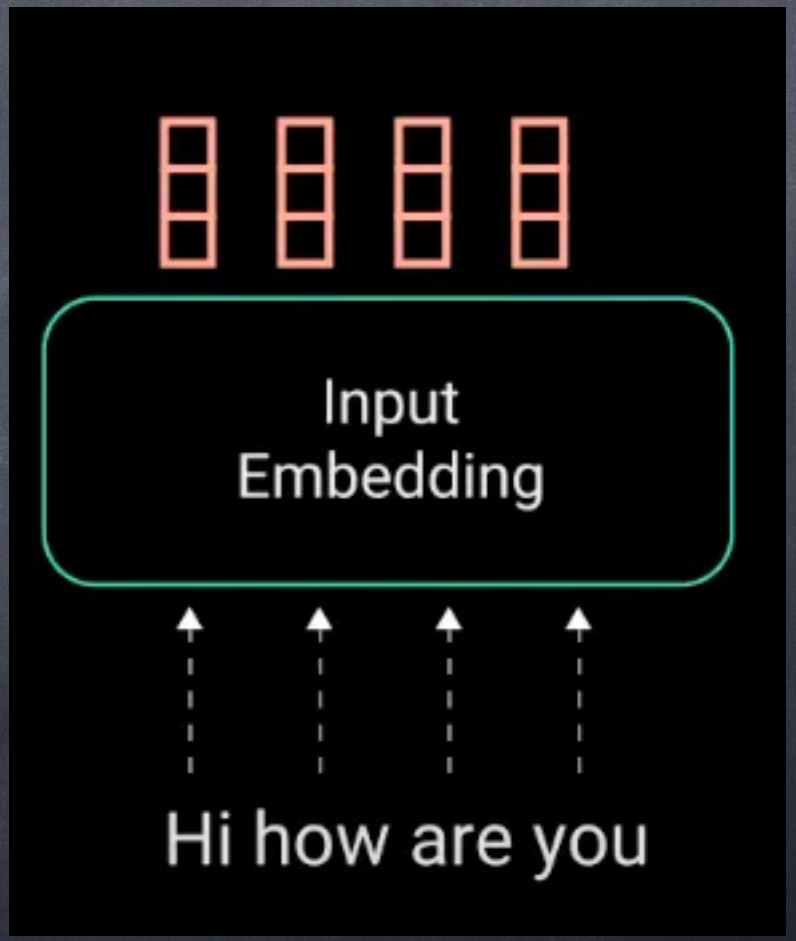
- Transformers (LSTMs are dead - long live Transformers)
- **Input:** “As Aliens entered our planet”.
- **Transformer output:** “and began to colonized Earth, a certain group of extraterrestrials began to manipulate our society through their influences of a certain number of the elite to keep and iron grip over the populace.”

- Transformer

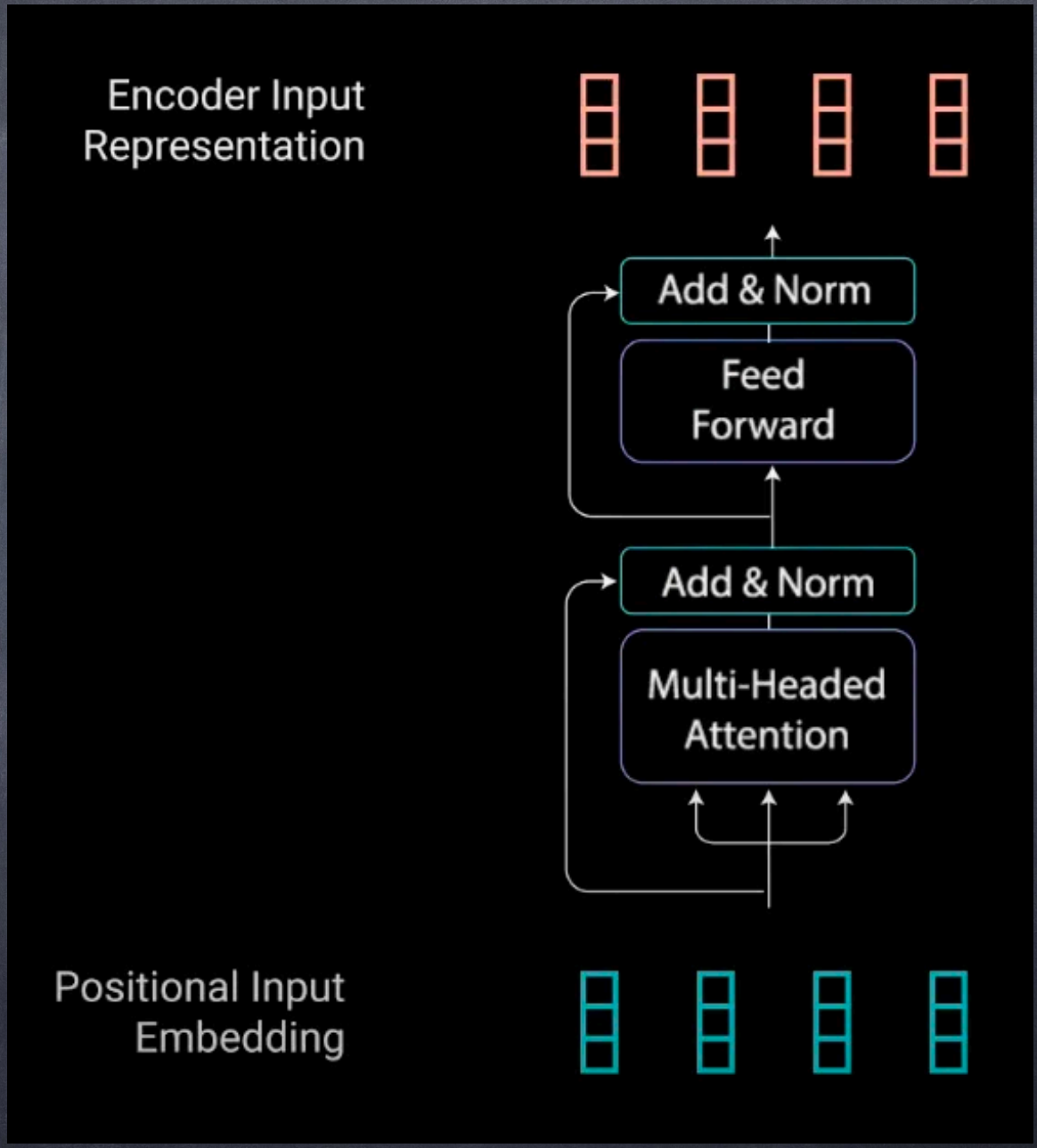


Input

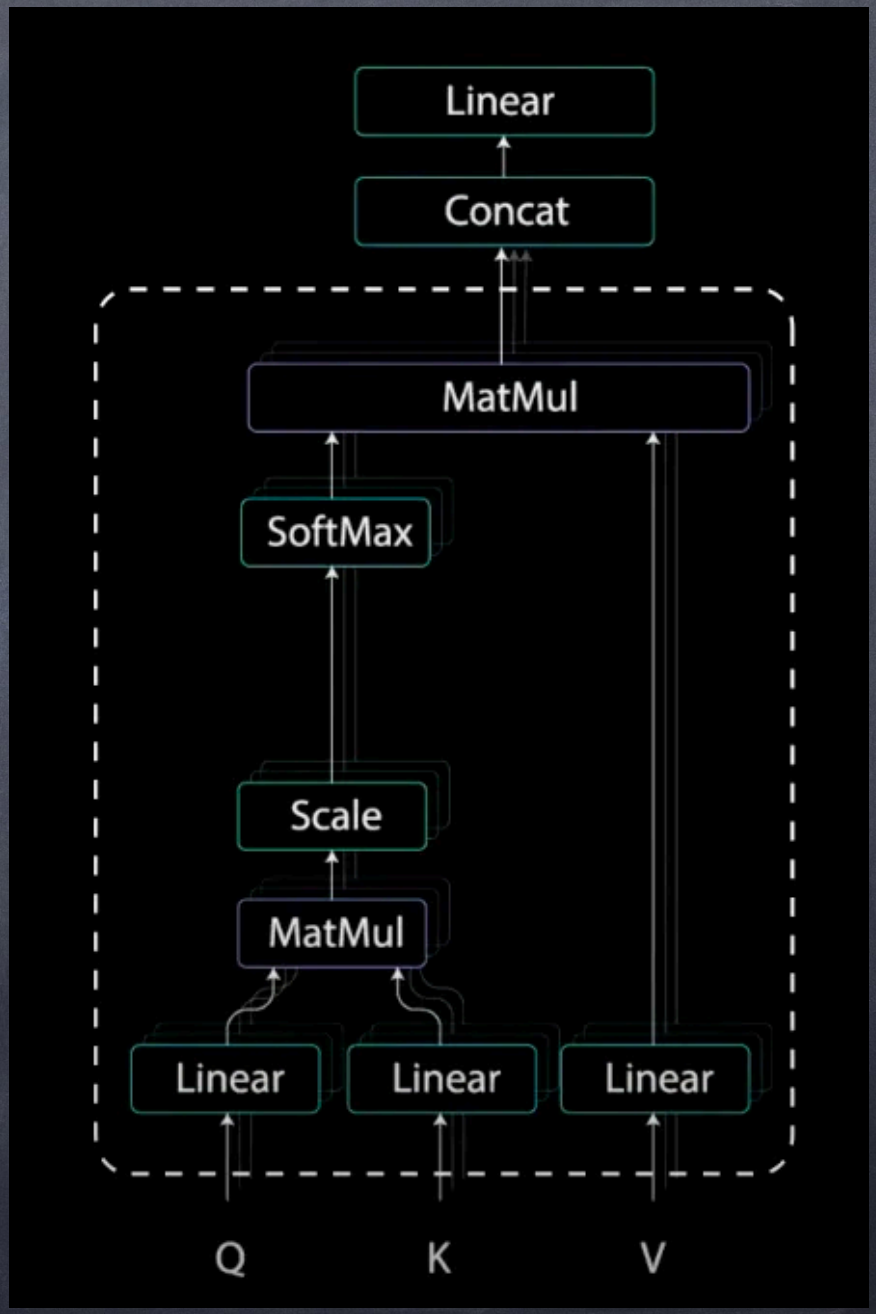
- Input embedding
- Positional encoding



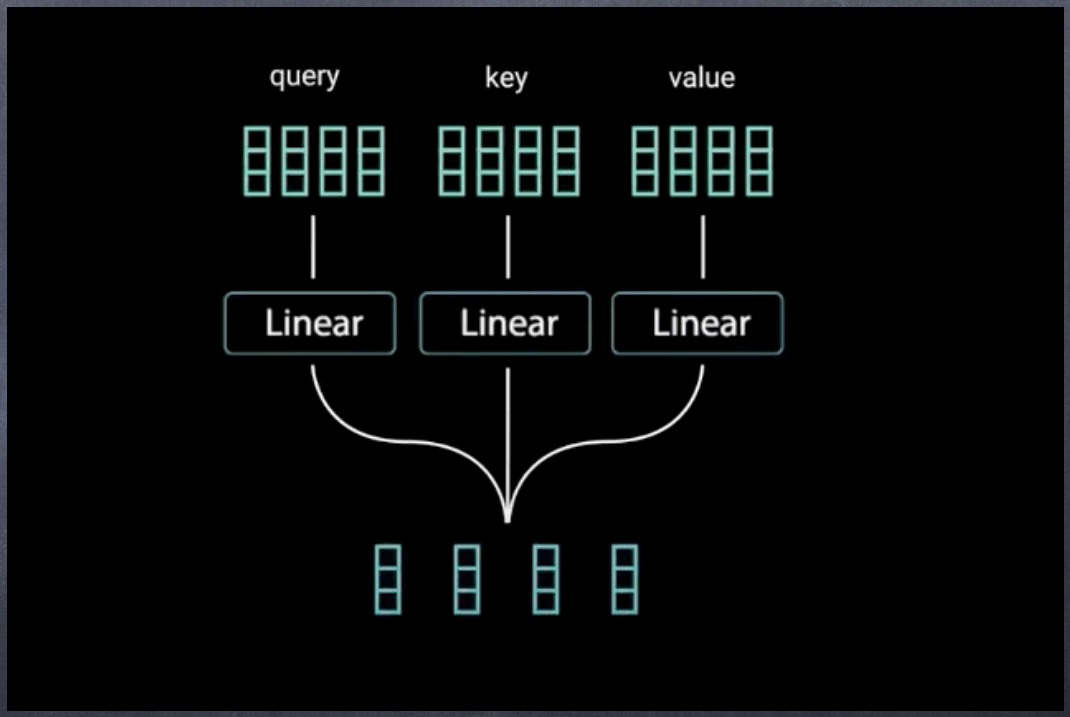
- Encoder
- multi-headed attention, followed by a fully connected network
- There are residual connections around each of the two sublayers followed by a layer normalization

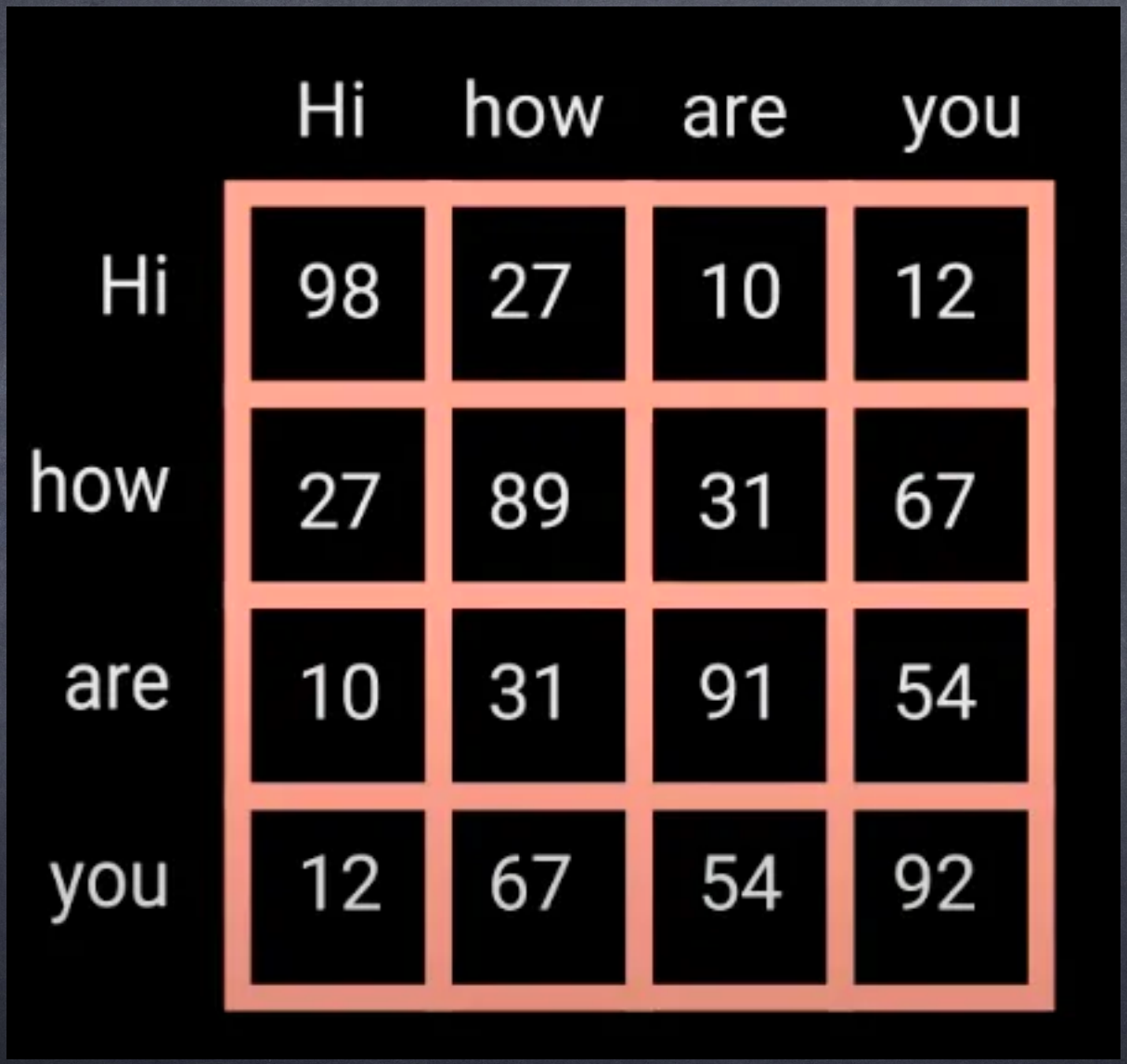


- Multi-Headed Attention
- Query, Key, and Value Vectors
- "The query key and value concept come from retrieval systems. For example, when you type a query to search for some video on Youtube, the search engine will map your query against a set of keys (video title, description etc.) associated with candidate videos in the database, then present you the best matched videos (values).

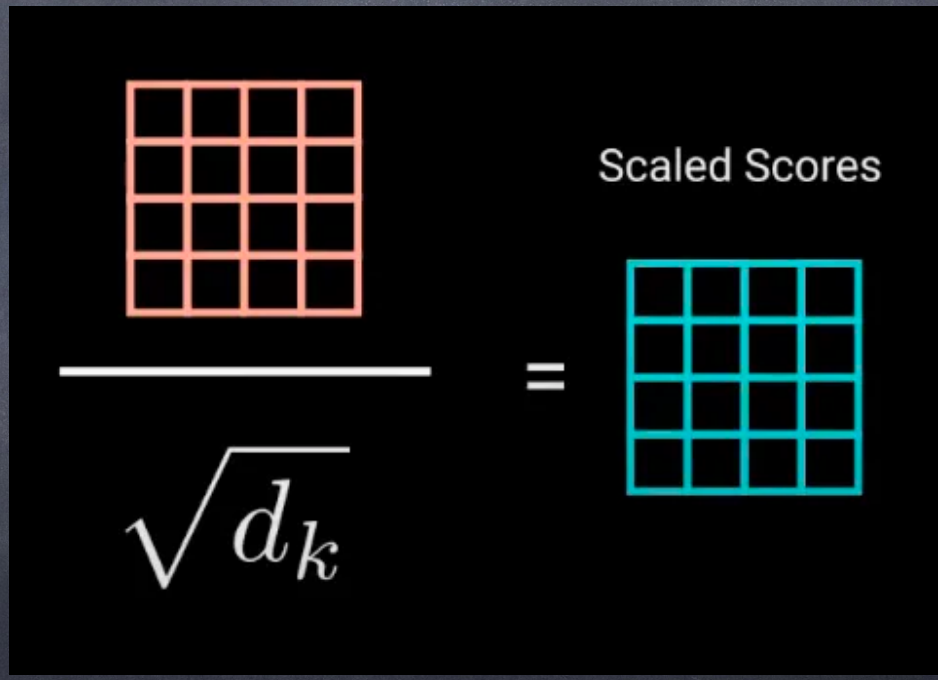


- Dot Product of Query and Key and score is obtained
- The score matrix determines how much focus should a word be put on other words. So each word will have a score that corresponds to other words in the time-step. The higher the score the more focus.

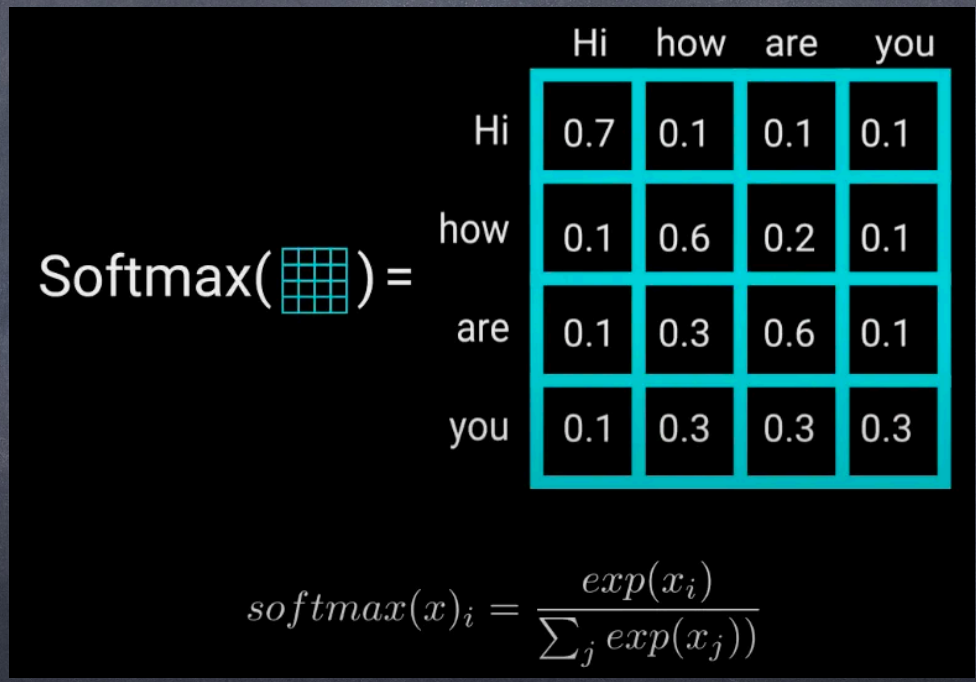



Hi
how
are
you
Hi
98
27
10
12
how
27
89
31
67
are
10
31
91
54
you
12
67
54
92

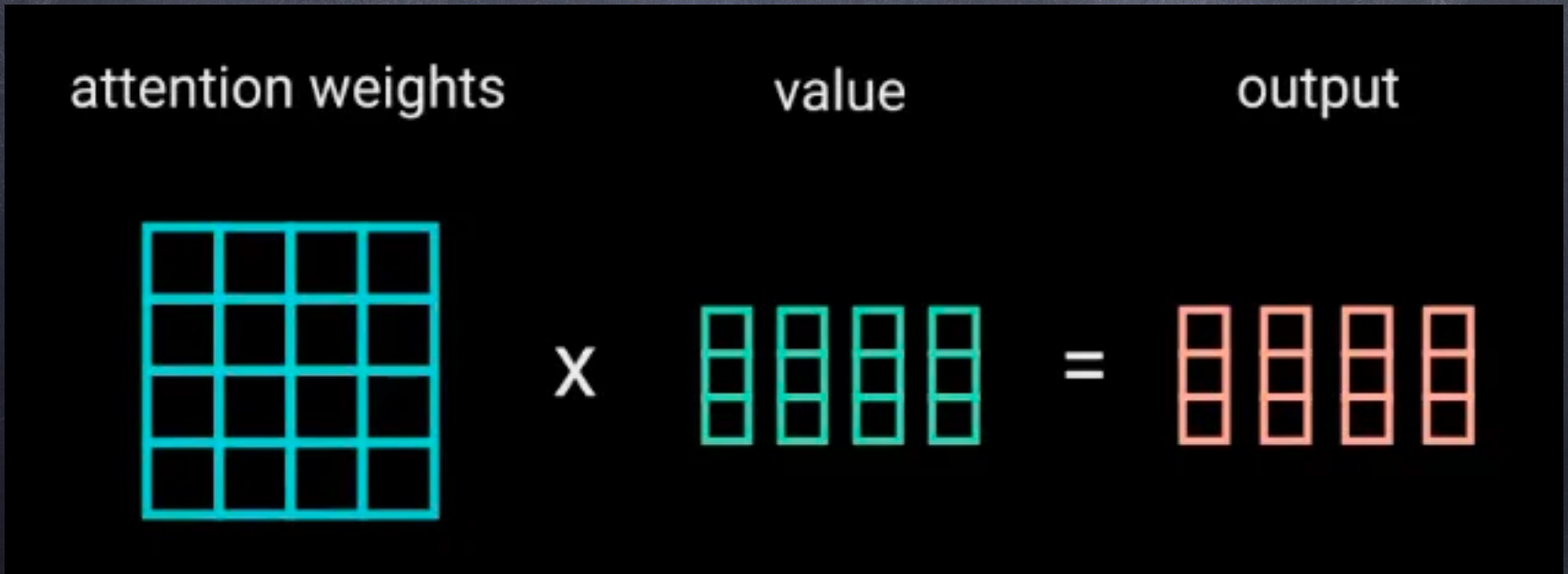
- Scaling Down the Attention Scores
- square root of the dimension of query and key



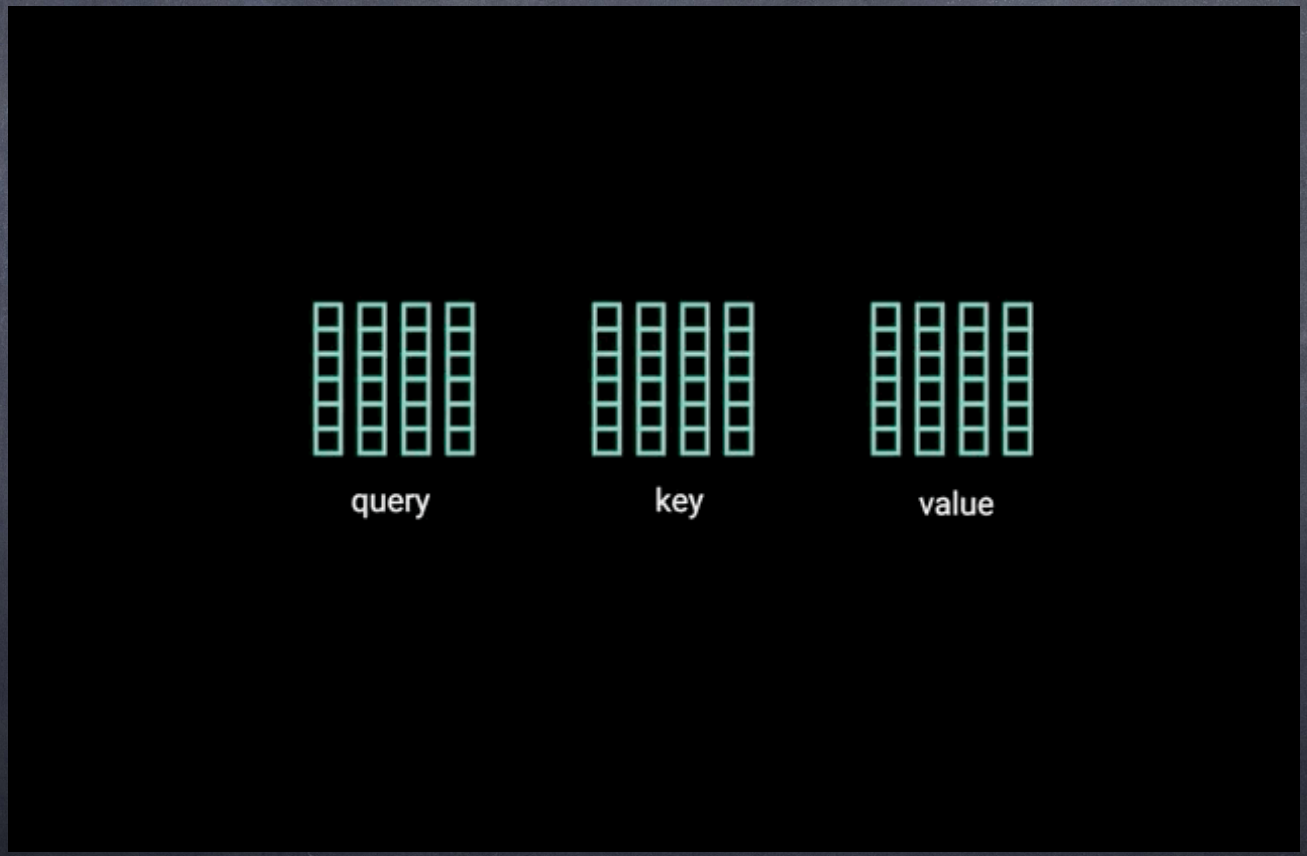
- Softmax of the Scaled Scores
- By doing a softmax the higher scores get heighten, and lower scores are depressed. This allows the model to be more confident about which words to attend too.



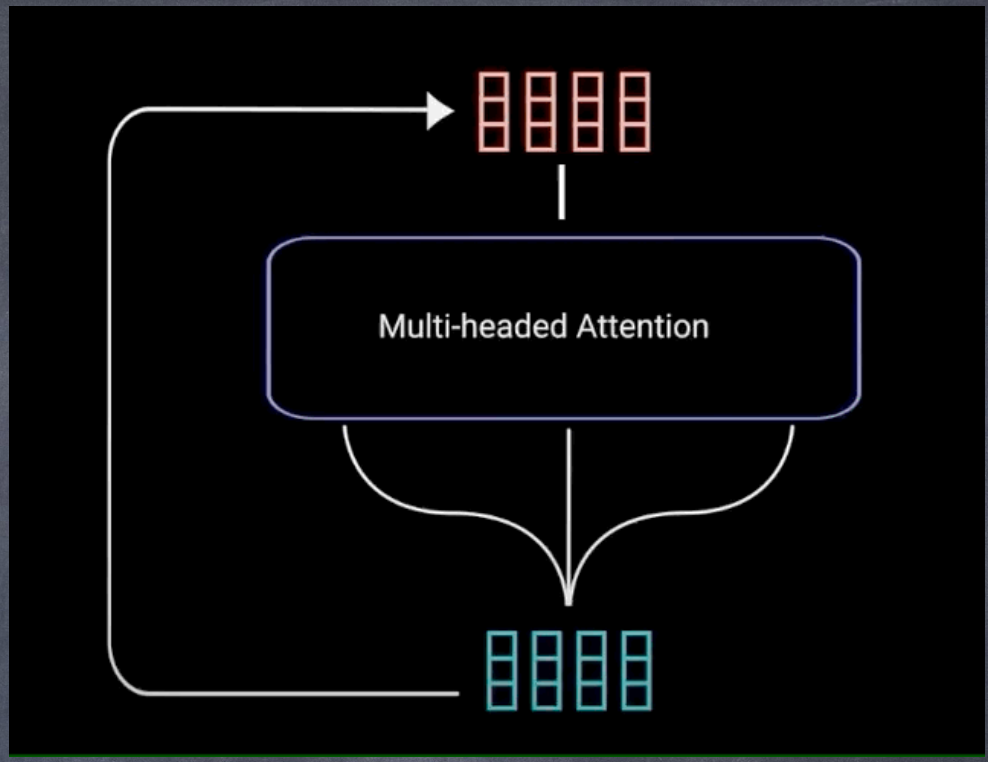
- Multiply Softmax Output with Value vector
- The higher softmax scores will keep the value of words the model learns is more important
- The lower scores will drown out the irrelevant words
- Then you feed the output of that into a linear layer to process.

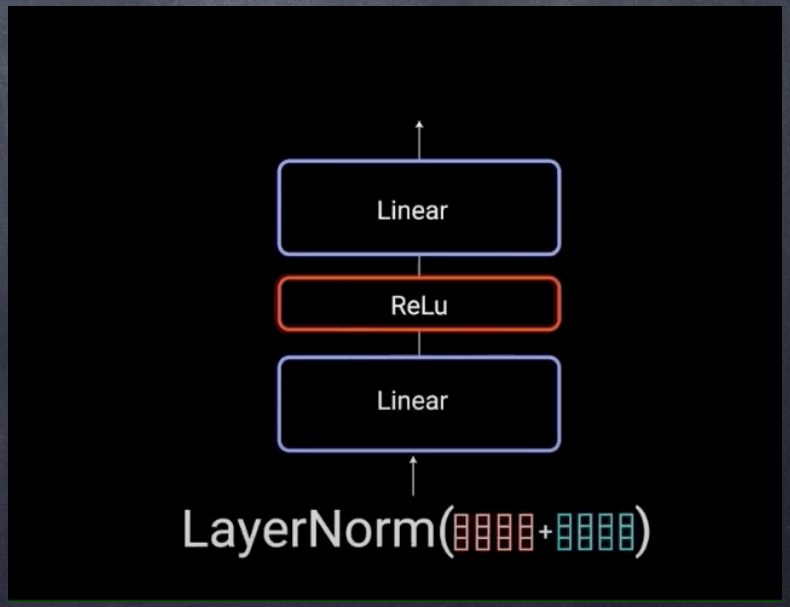


- Computing Multi-headed Attention

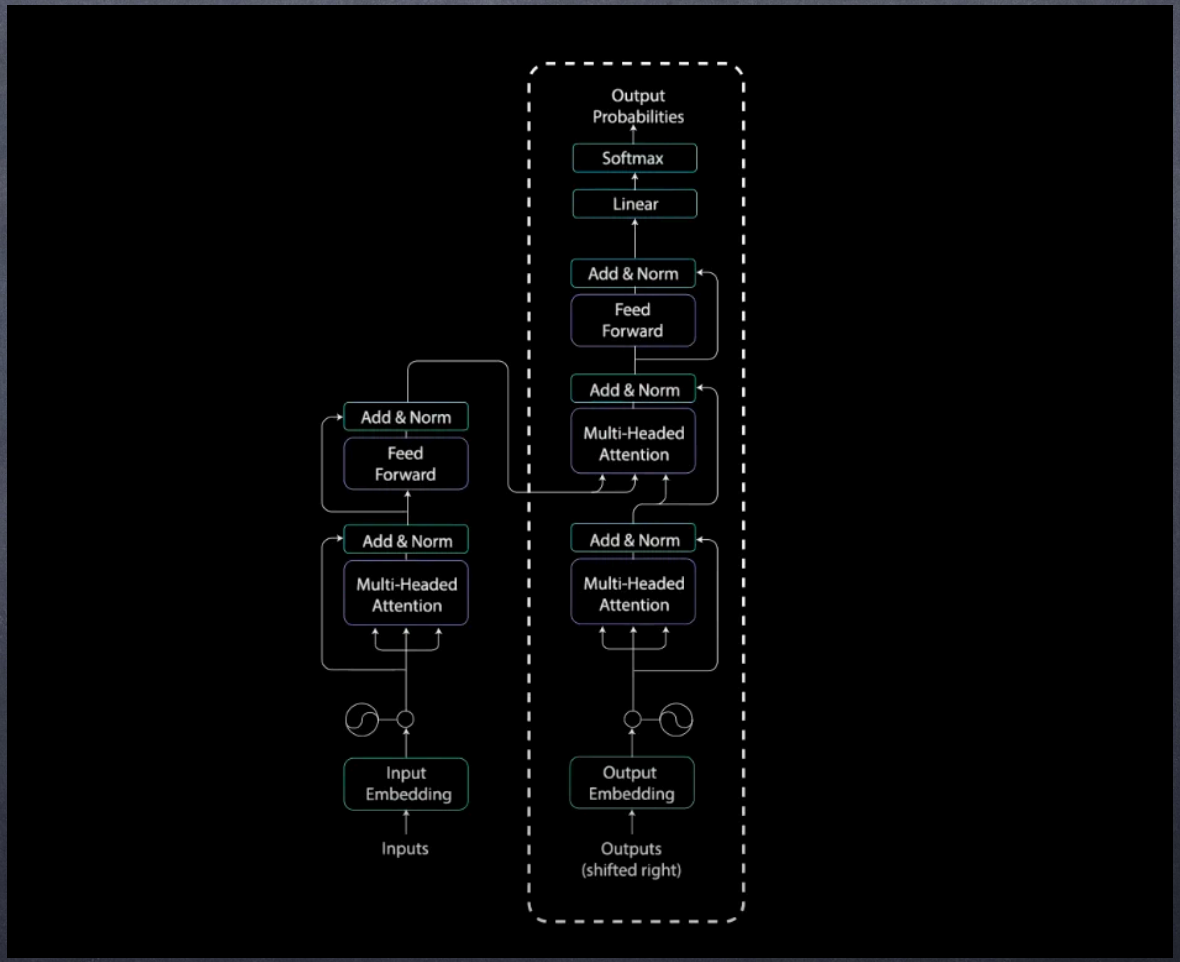


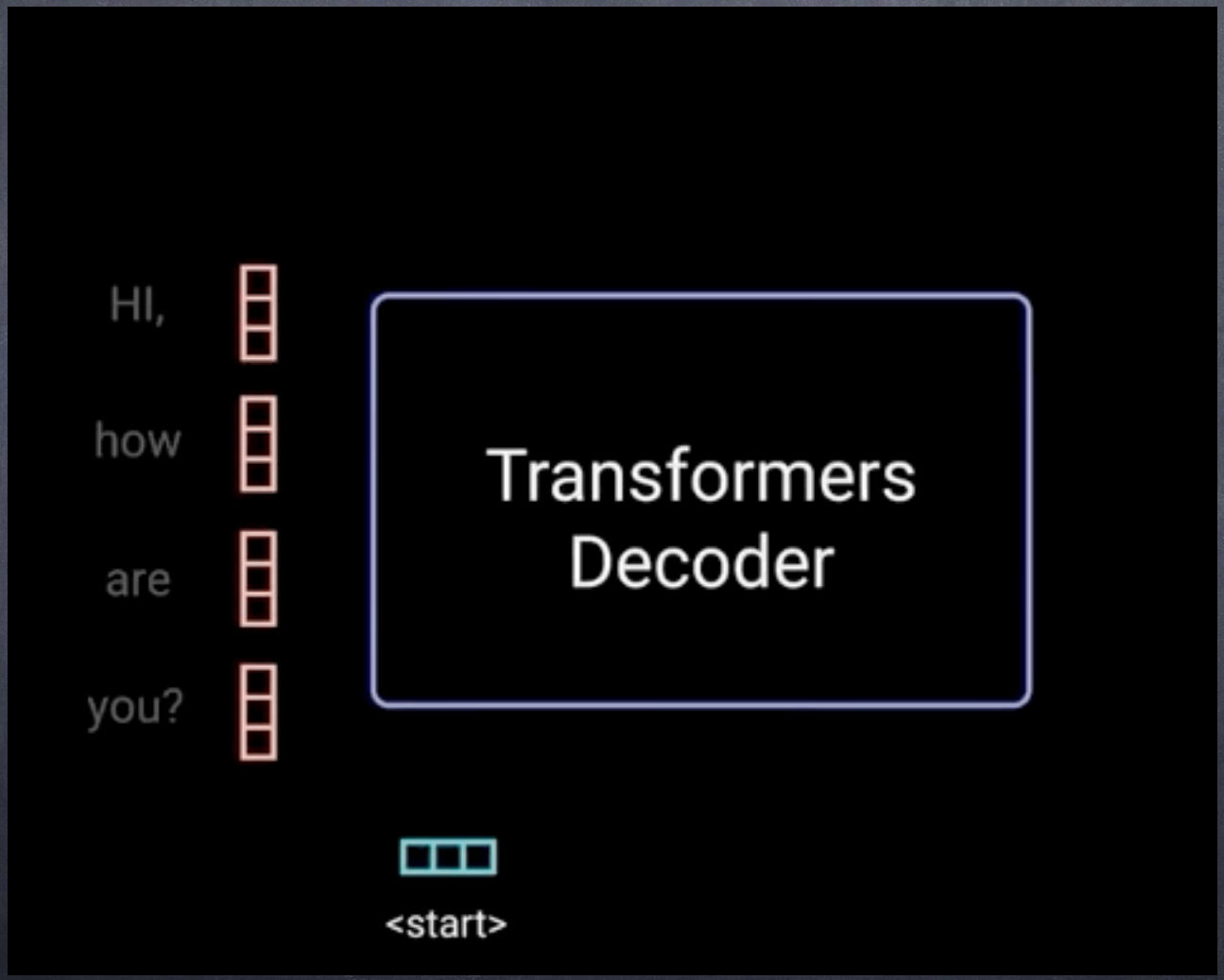
- The Residual Connections, Layer Normalization, and Feed Forward Network



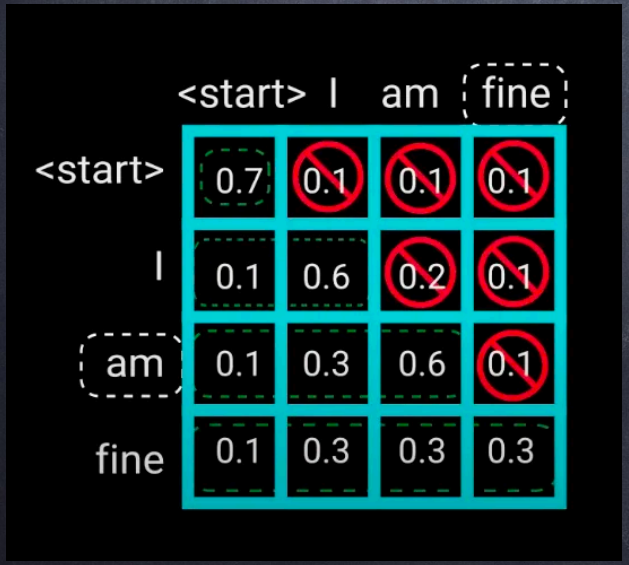


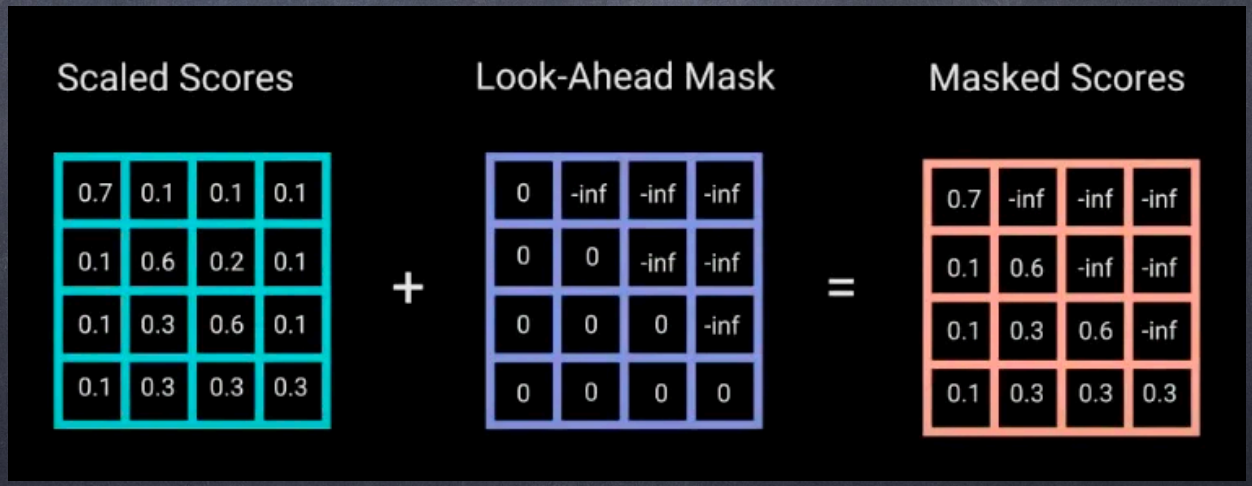
- Decoder

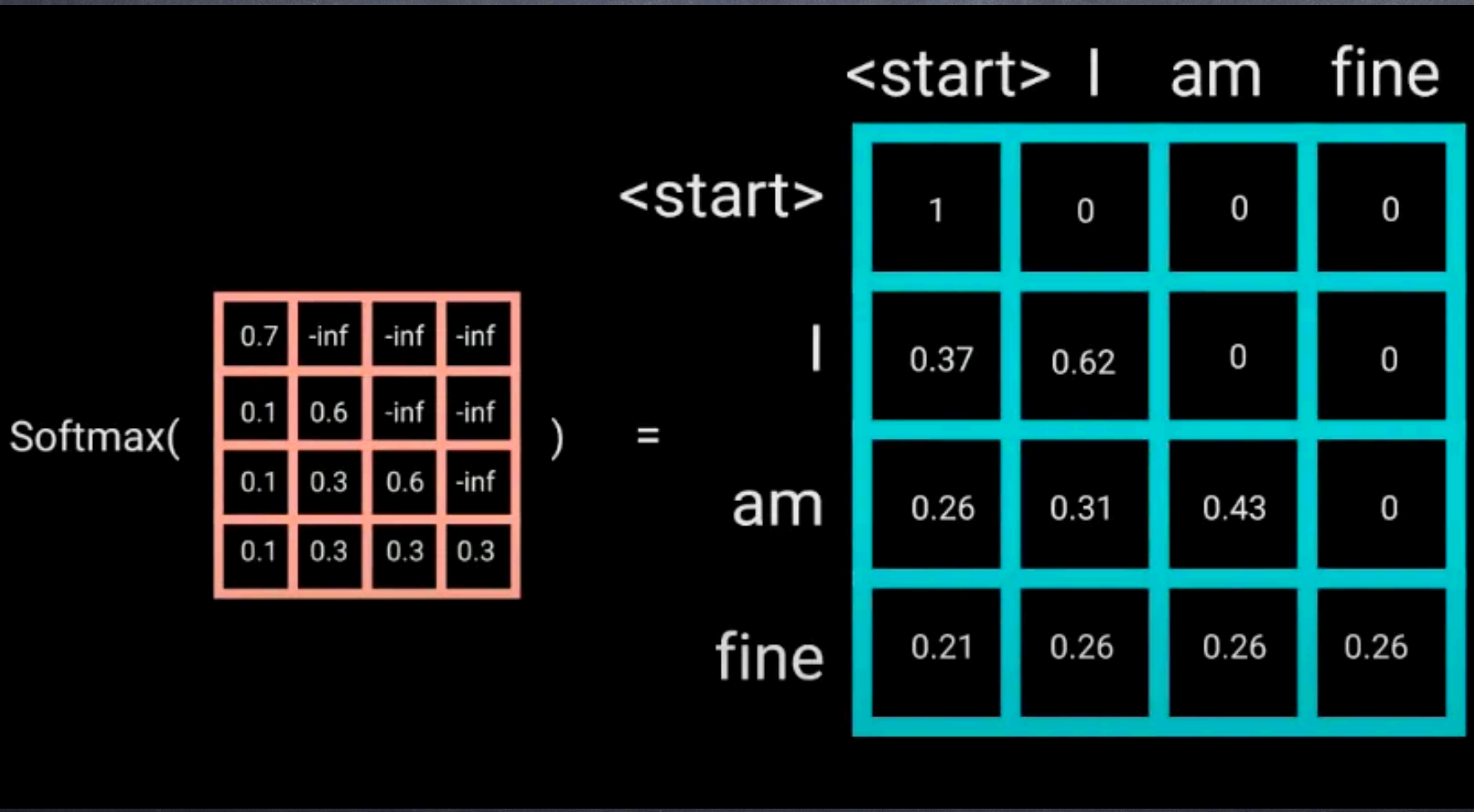

Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Add & Norm
Multi-Headed Attention
Add & Norm
Feed Forward
Add & Norm
Multi-Headed Attention
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

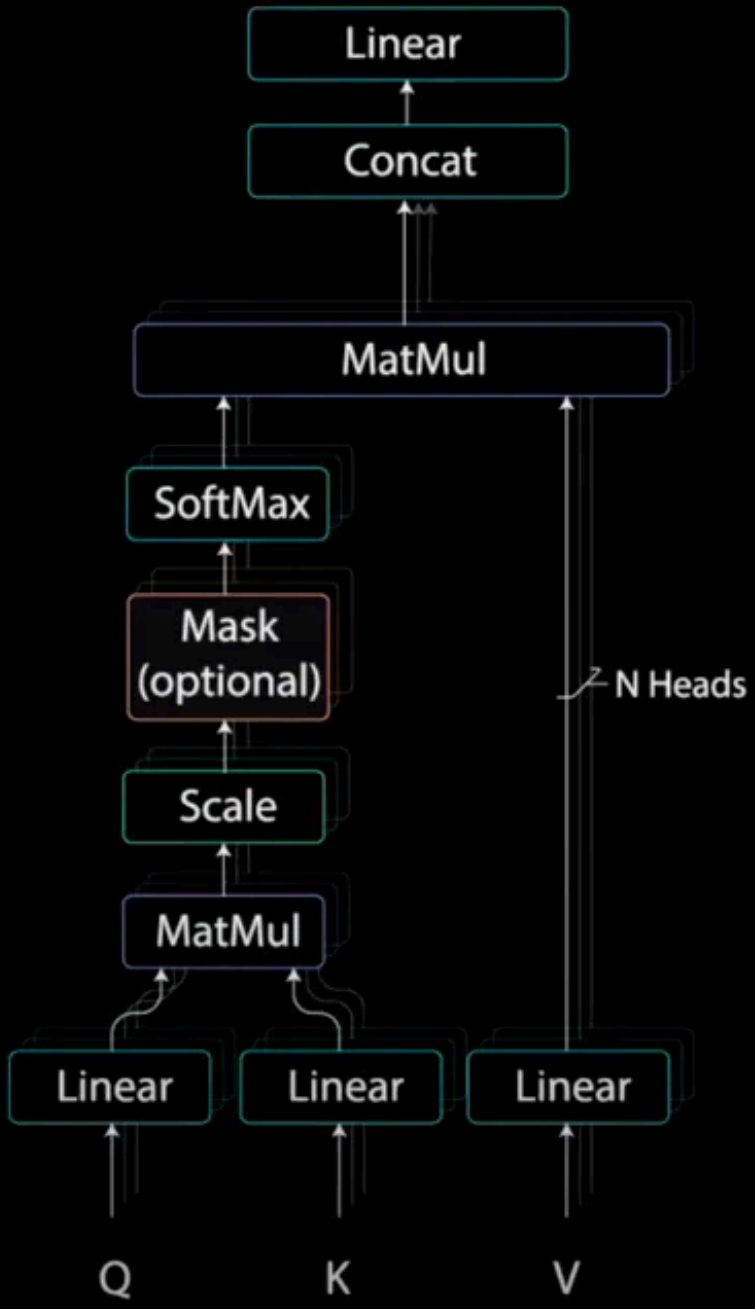

HI,
how
are
you?
Transformers Decoder
<start>

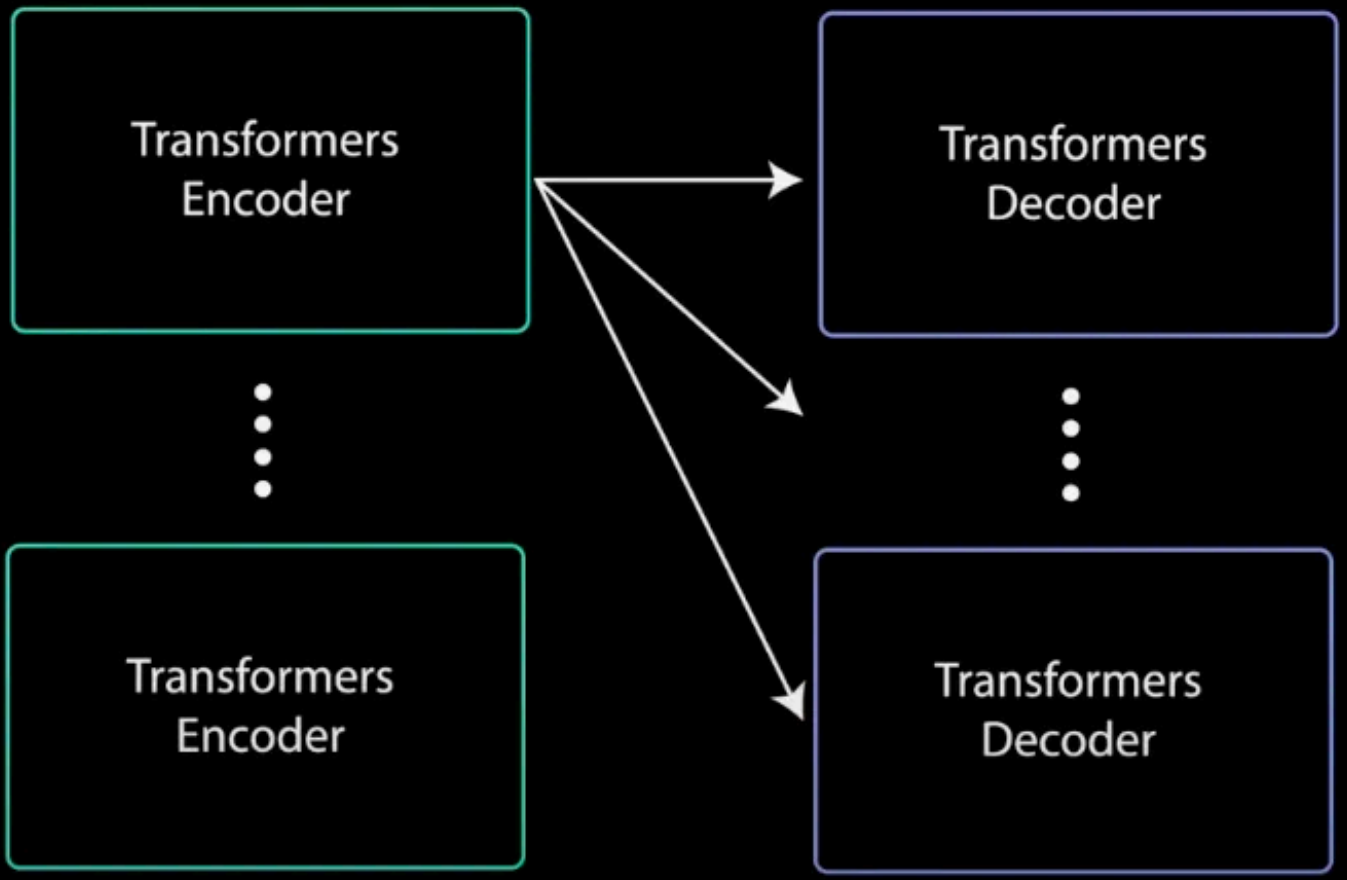
- Decoder Input Embeddings & Positional Encoding
- Decoders First Multi-Headed Attention






Softmax(
0.7 -inf -inf -inf
0.1 0.6 -inf -inf
0.1 0.3 0.6 -inf
0.1 0.3 0.3 0.3
)
=
<start> I am fine
<start> 1 0 0 0
I 0.37 0.62 0 0
am 0.26 0.31 0.43 0
fine 0.21 0.26 0.26 0.26


Linear
Concat
MatMul
SoftMax
Mask
(optional)
Scale
MatMul
N Heads
Linear
Linear
Linear
Q
K
V


Transformers
Encoder
Transformers
Encoder
Transformers
Decoder
Transformers
Decoder

And That's it ...

- Next class: Multi-task learning and Transfer Learning