Attention mechanism

Motivation

- LSTMs help with memory, but it's still hard.
- For encoder/decoder models, the fixed-length embedding is problematic.

we can choose the output of decoder to be large, but still finite

Encoder-decoder

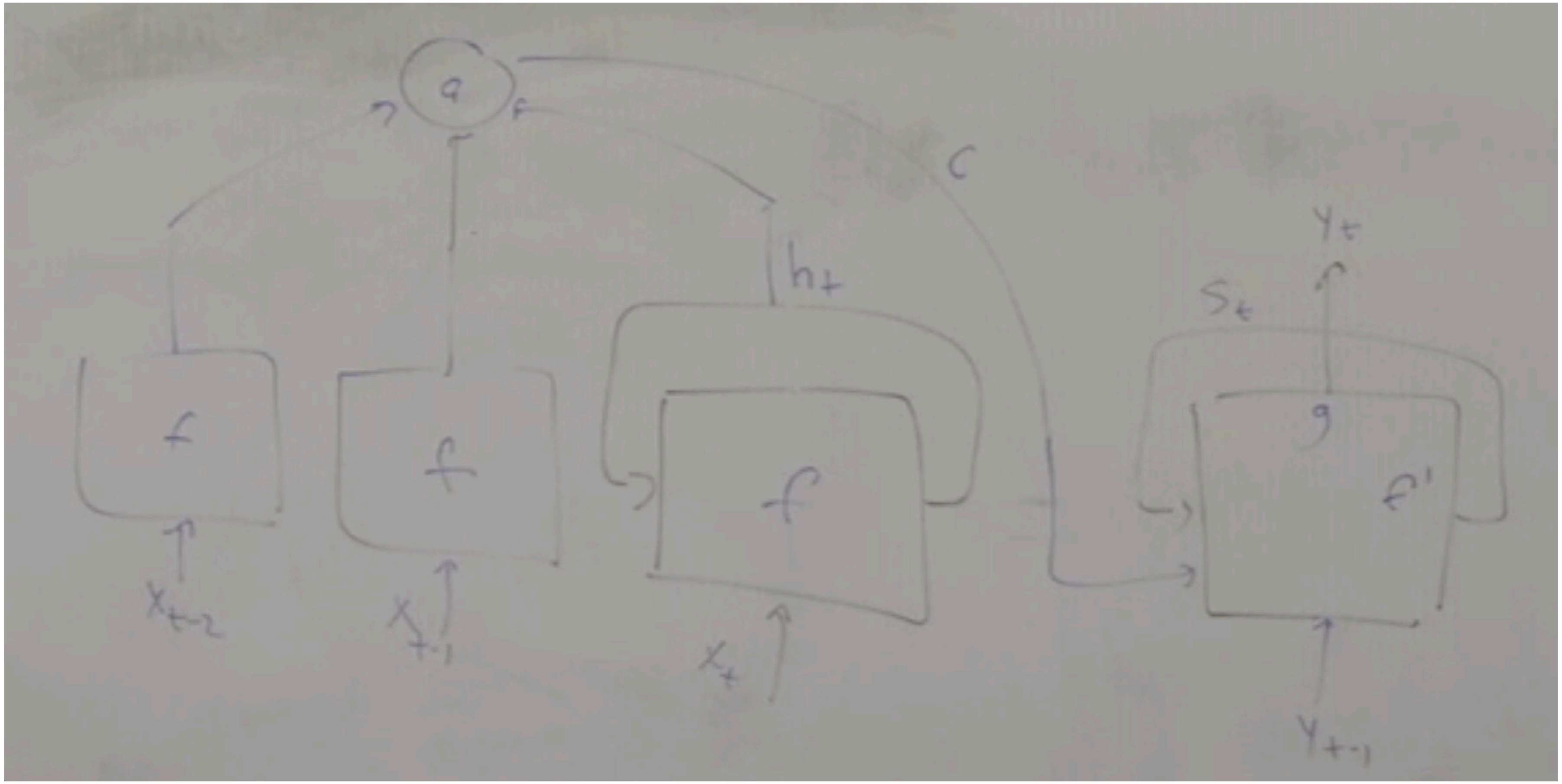
$$h_t = f(x_t, h_{t-1})$$

q— somehow aggregate all of the ht (mean, median)

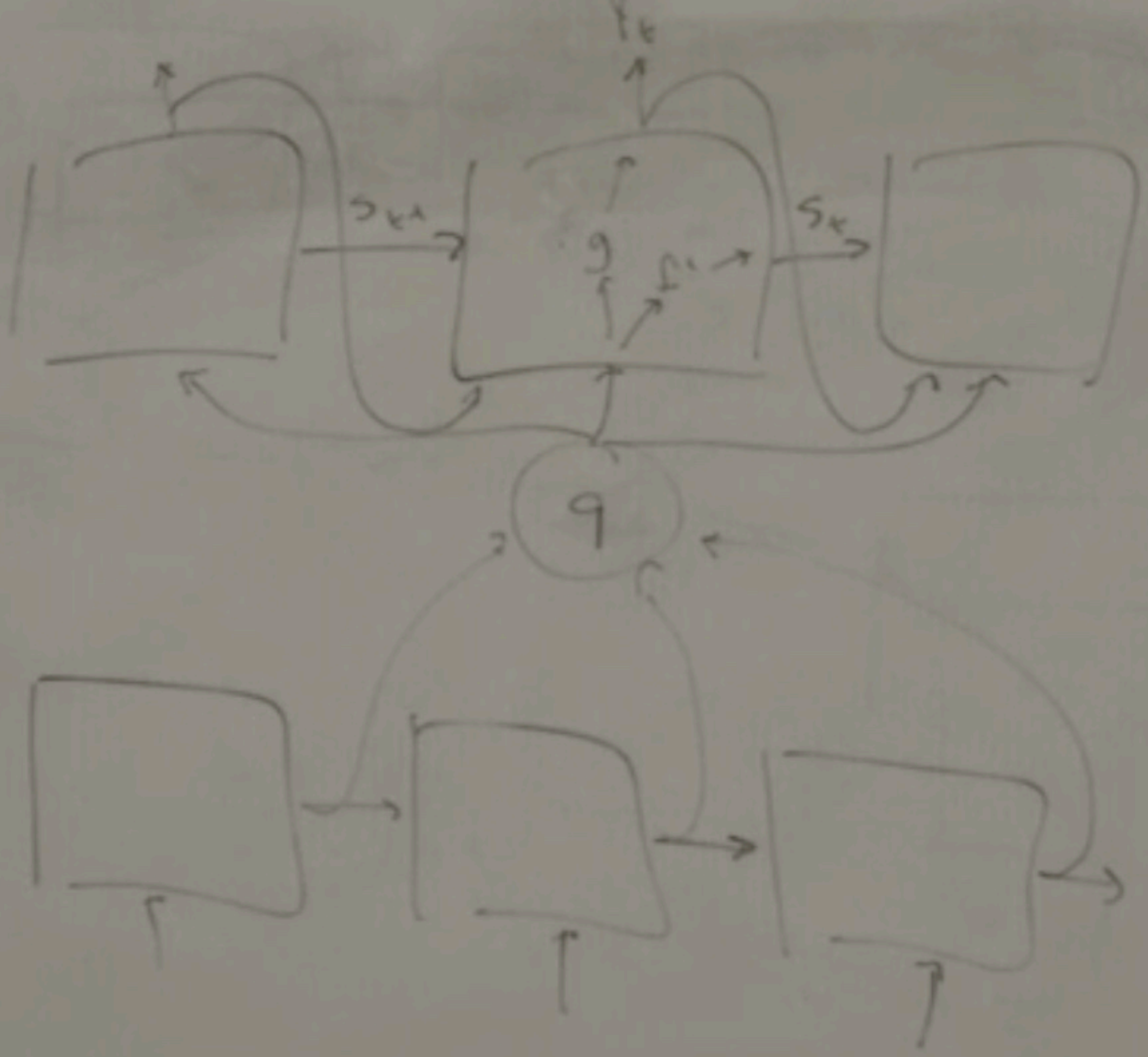
$$c = q(\{h_1, ..., h_{T_x}\})$$

$$p(y_t | \{y_1, ..., y_{t-1}\}, c) = g(y_{t-1}, s_t, c)$$

$$s_t = f'(s_{t-1}, y_{t-1}, c)$$



a
c
h_t
S_t
Y_t
f
f
f
g
f'
X_{t-2}
X_{t-1}
X_t
Y_{t-1}



y_t
s_{t-1}
s_t
g
f'
q

with attention

decoder： state output

$$s_i = f'(s_{i-1}, y_{i-1}, c_i)$$

compute a special weighted
sum of the input information

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

encoder： hidden state

aij describe the relationship between output i and input j

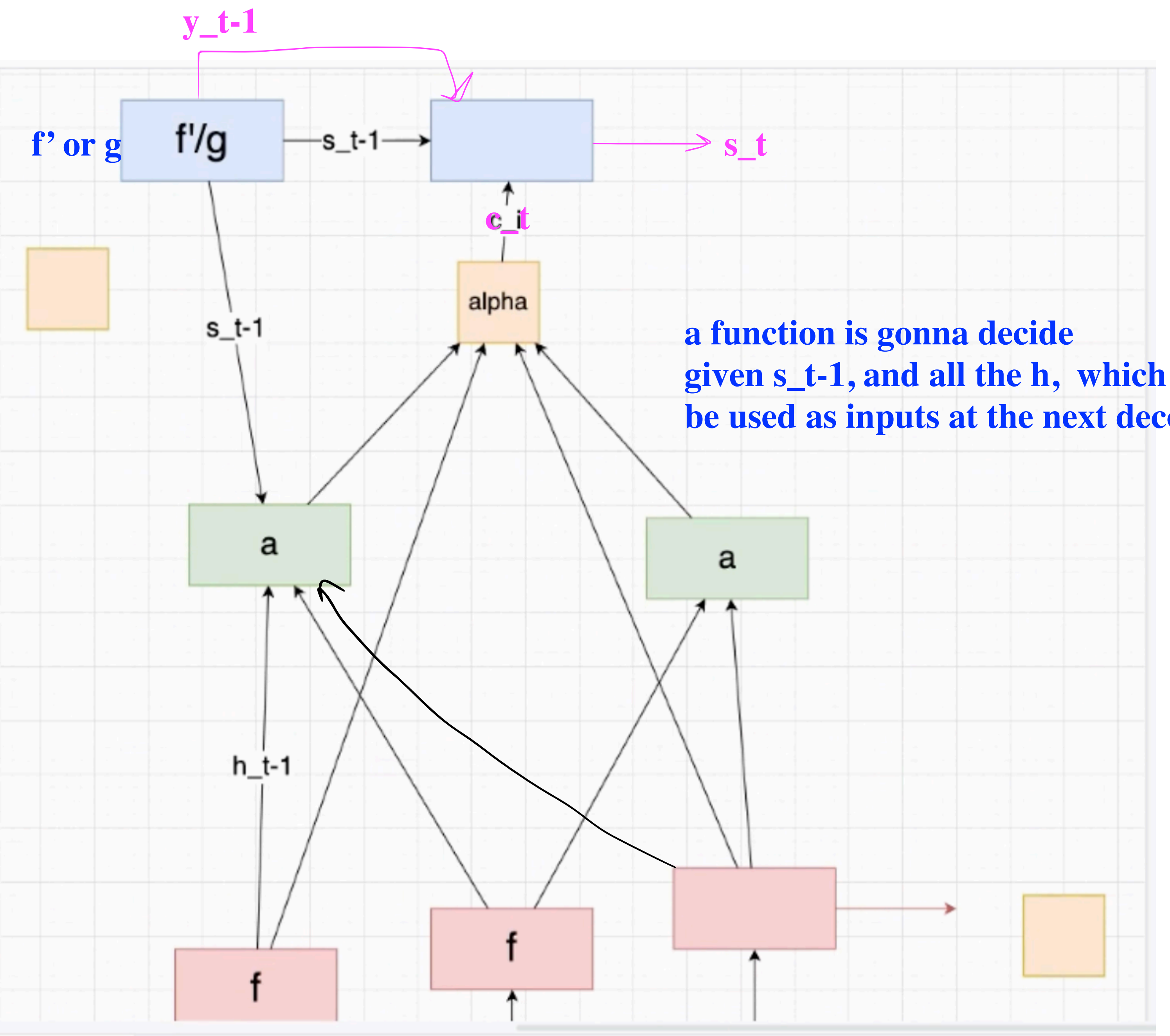
i for decoder, j for encoder

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

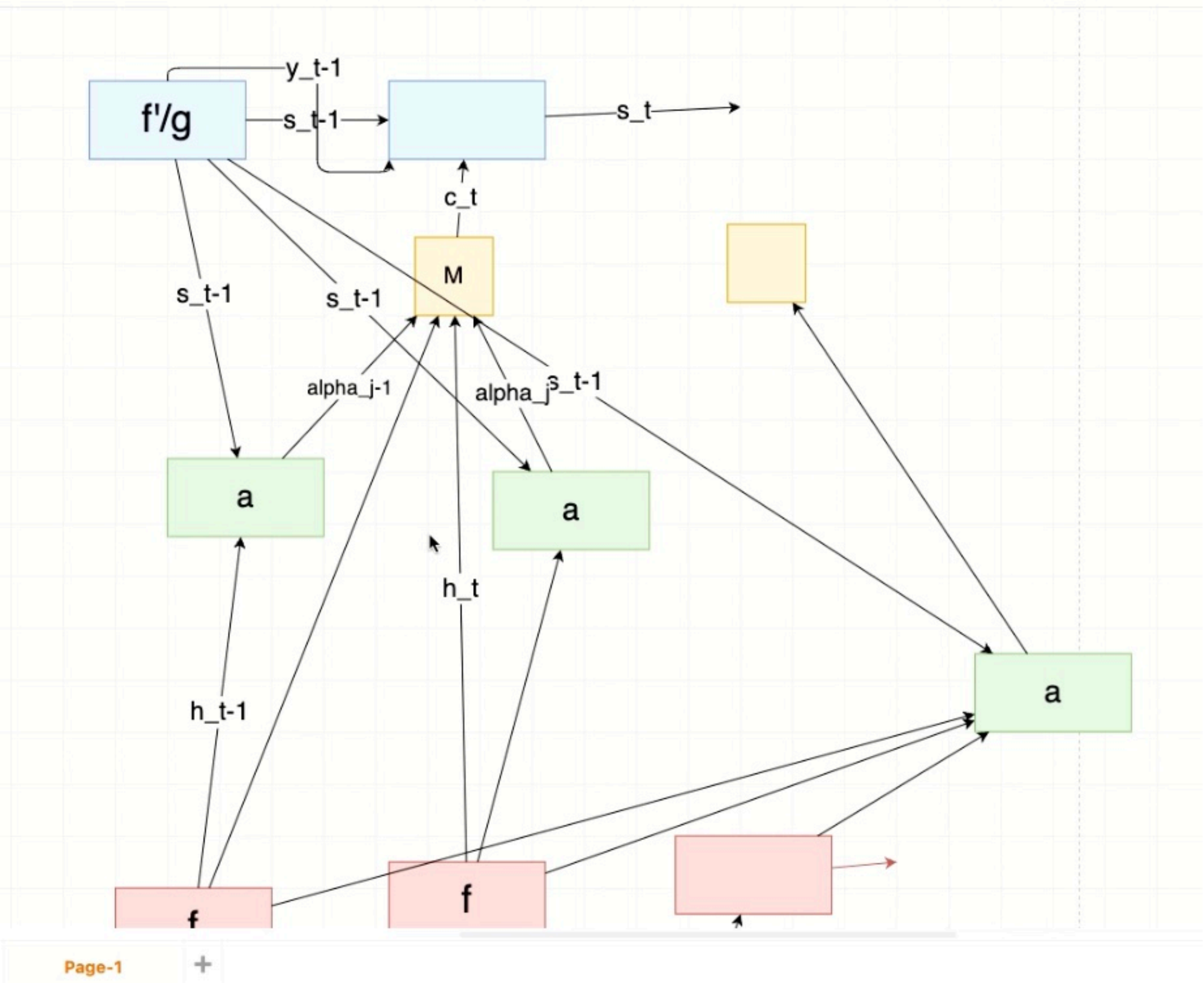
just normalizng so that sum to 1

$$e_{ij} = a(s_{i-1}, h_j)$$

a是一个公式 QKt/sqrt（n）



a function is gonna decide
given s_t-1, and all the h, which one should
be used as inputs at the next decoder time set

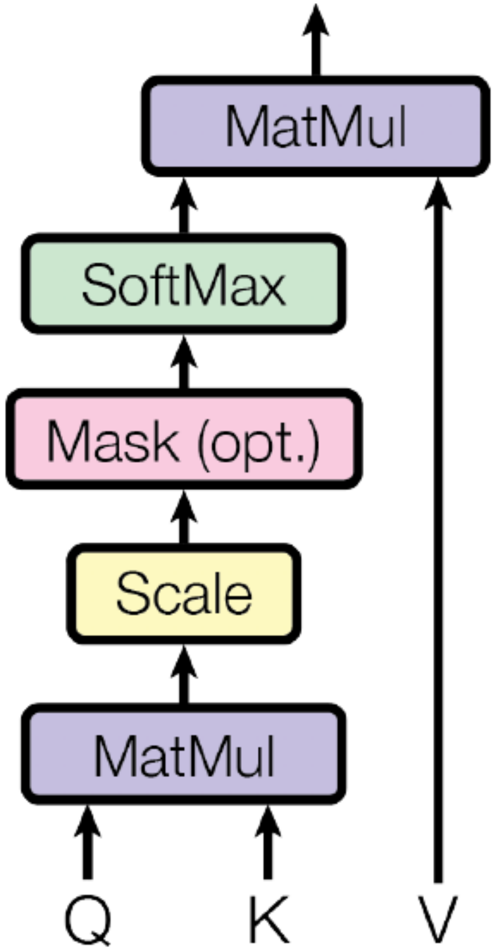


f'/g
y_t-1
s_t-1
s_t
c_t
M
s_t-1
s_t-1
alpha_j-1
alpha_j
s_t-1
a
a
h_t
a
h_t-1
f
f
Page-1

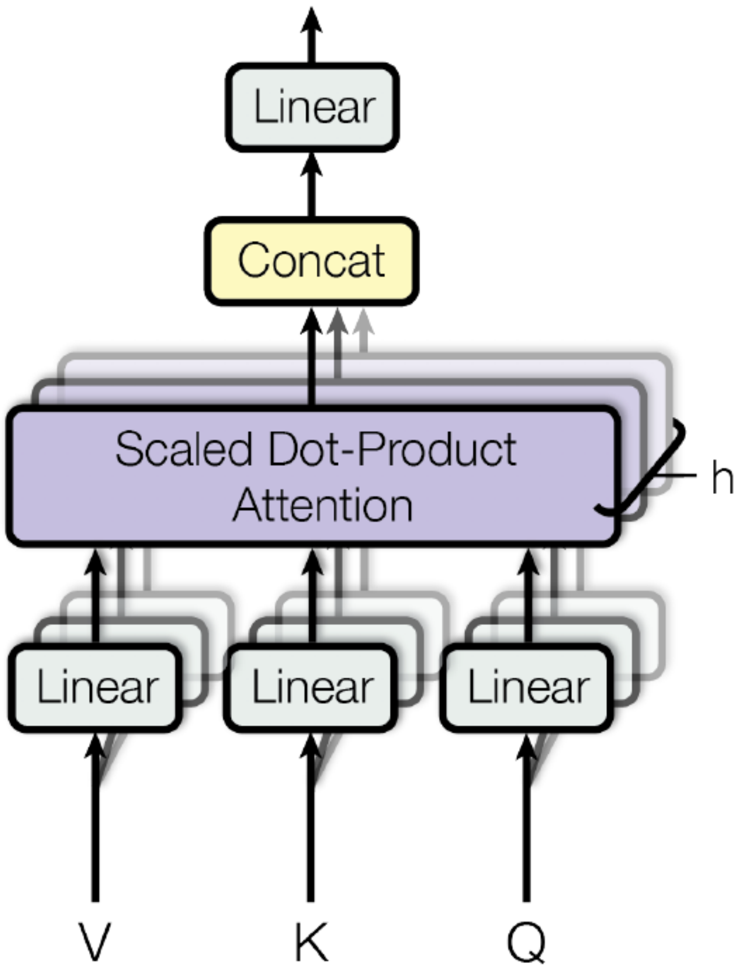
Broad idea of attention:
we compute how much to care about each input, specially given each output (with real information about that output which comes from the previous state(s_t-1)

this did two things for us when we sort of reduced that bottleneck of the big context sector by saying that we get compute the context independently for each output
and it shortens the network (the shortest path between each output and each input has shrunk.

the output just need to decide to depend on that input and the path (the number of weights and activation function that information need to go through to get where it needs to go.


MatMul
SoftMax
Mask (opt.)
Scale
MatMul
Q
K
V

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{n}}\right)\mathbf{V}$$


Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
K
Q

Resources

- Neural Machine Translation by Jointly Learning to Align and Translate
- Attention Is All You Need