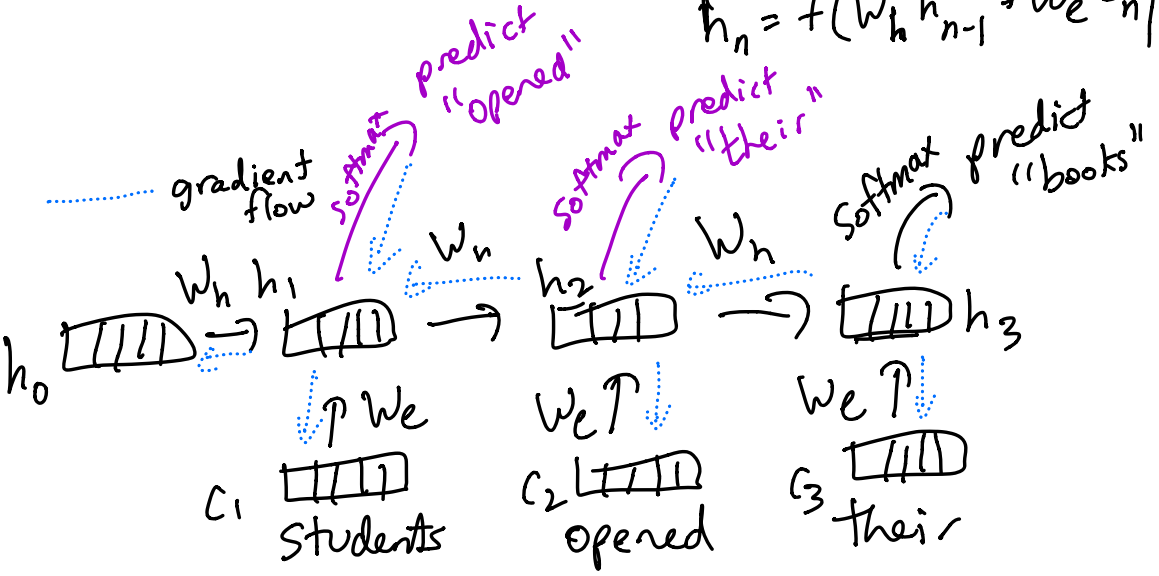
Recurrent neural network:

$$h_n = f(W_h h_{n-1} + W_e c_n)$$



RNN gradient ⇒ "backpropagation thru time"

$$L_3 = -\log P(\text{books} \mid \text{"students opened their"})$$

$$L_2 = -\log P(\text{their} \mid \text{"students opened"})$$

$$L_1 = -\log P(\text{opened} \mid \text{students})$$

$$L = \frac{L_1 + L_2 + L_3}{3}$$

} average neg. log likelihood of the ground-truth next word over all tokens in the batch

batch:

1. students opened their books
2. people walked their dog
3. the classroom fell silent

Issues w/ RNNs:

1. "bottleneck"
   - Ray Mooney, 2014
     "you can't represent the meaning of a sentence in a BLEEPING vector!"
2. lack of parallelism across timesteps
   - I can only compute h_n after computing h_{n-1}

---

attention mechanism:

history: - developed initially for RNNs and for machine translation
- Bahdanau, Cho et al, 2014
- Vaswani et al. 2017 dropped the "recurrent" aspect and created a fully attention based architecture
  - Transformer
  - for machine translation
  - introduced by Google
  - hidden state at each timestep is independent of $h_{1 \ldots n-1}$

self-attention: parallel at training time, sequential at test-time

Self-attention:

computation of hidden state at time step 3:

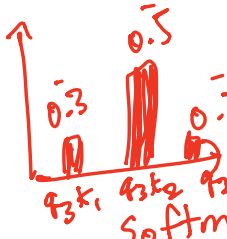
$h_3 = 0.3v_1 + 0.5v_2 + 0.2v_3$

↓

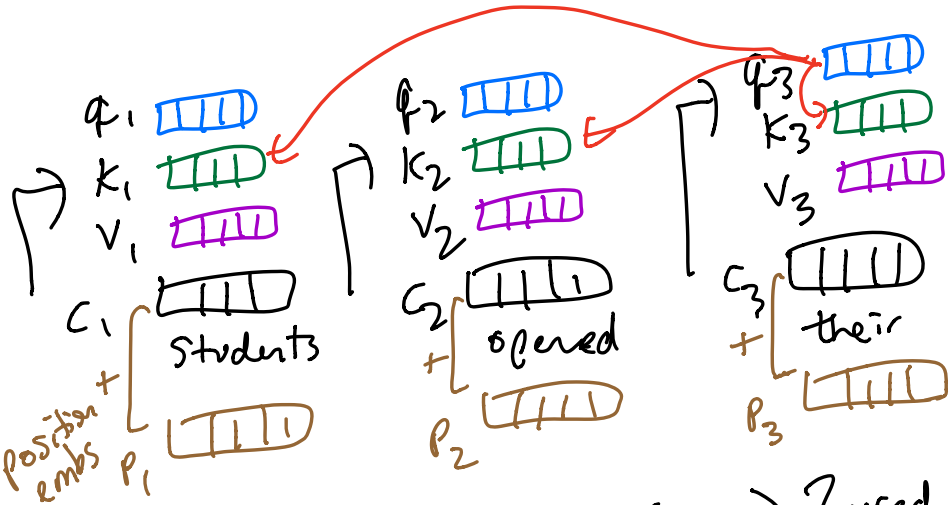
→ Softmax

↓

predict "books"



attn scores: $\langle q_3 \cdot k_1, q_3 \cdot k_2, q_3 \cdot k_3 \rangle$



query: $q_1 = f(W_q c_1)$ $q_2 = f(W_q c_2)$ } used to compute "attention scores"

key: $k_1 = f(W_k c_1)$

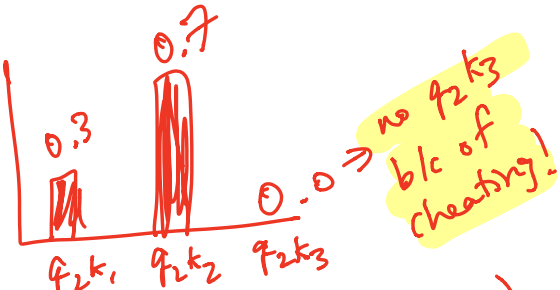
value: $v_1 = f(W_v c_1)$ } encode information used to compute hidden state

W_q, W_k, W_v are randomly initialized parameters learned during training!

computations @ second time step:

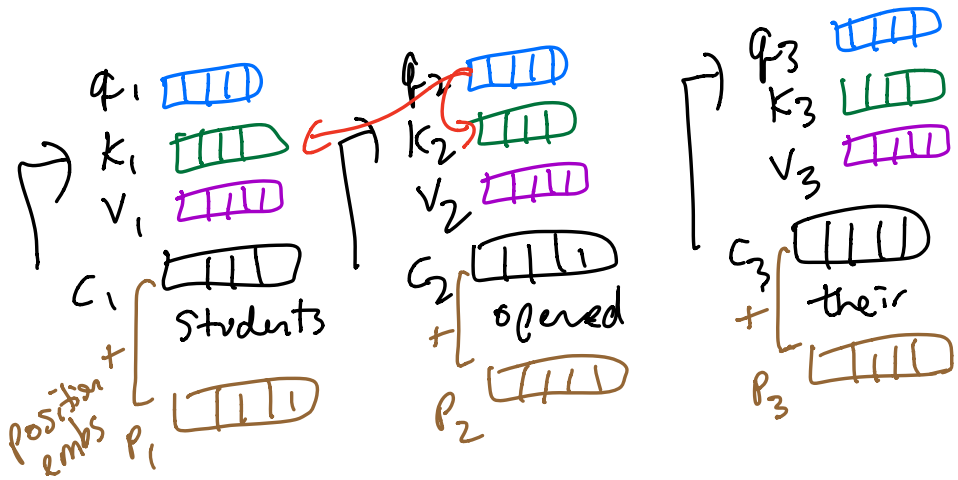
$h_2 = 0.3 \cdot v_1 + 0.7 \cdot v_2$

↳ Softmax layer → predict "their"



$\text{Softmax}(W_o h_2)$

attn: $\text{Softmax}(\langle q_2 \cdot k_1, q_2 \cdot k_2 \rangle)$



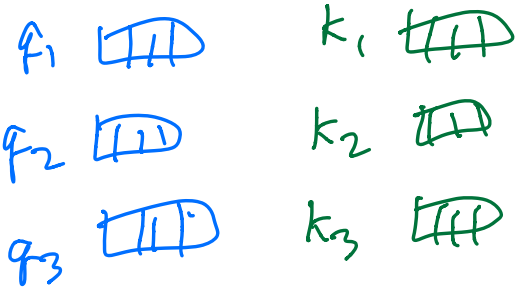
---

no dependencies between h_1, h_2, h_3

↳ parallelize

↳ reduce bottleneck

how to parallelize:

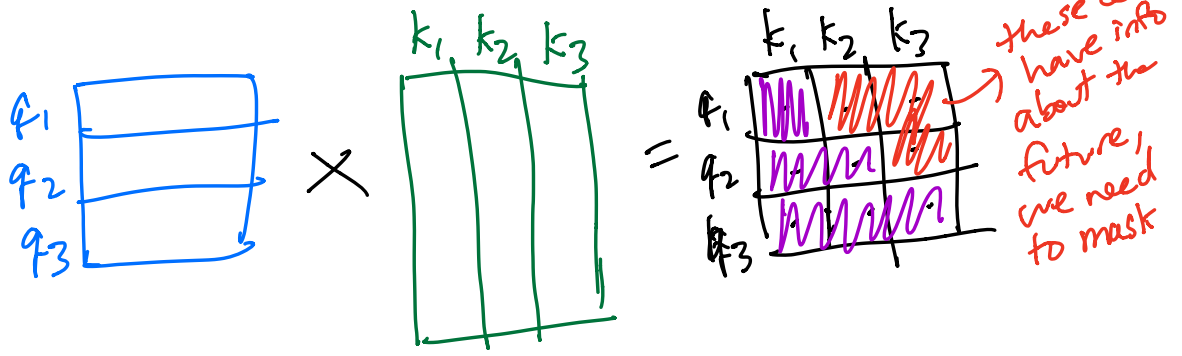


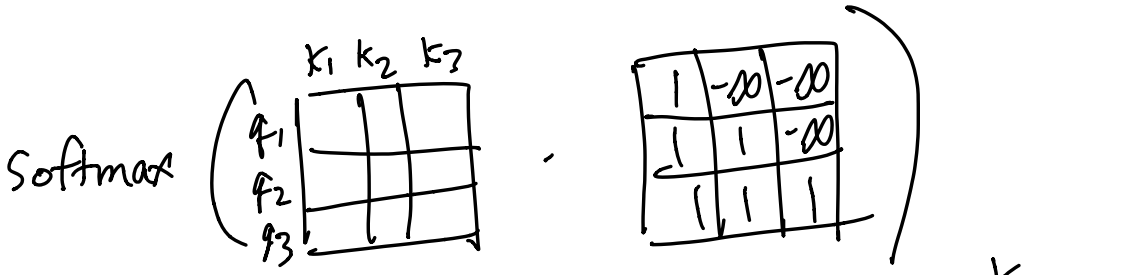
attn vectors

$a_1 = \langle q_1 k_1 \rangle$

$a_2 = \langle q_2 k_1, q_2 k_2 \rangle$

$a_3 = \langle q_3 k_1, q_3 k_2, q_3 k_3 \rangle$





attn scores

mask matrix