Sequence Encoding
Basic Attention

國立臺灣大學 資訊工程學系
陳縕儂 助理教授
http://vivianchen.idv.tw

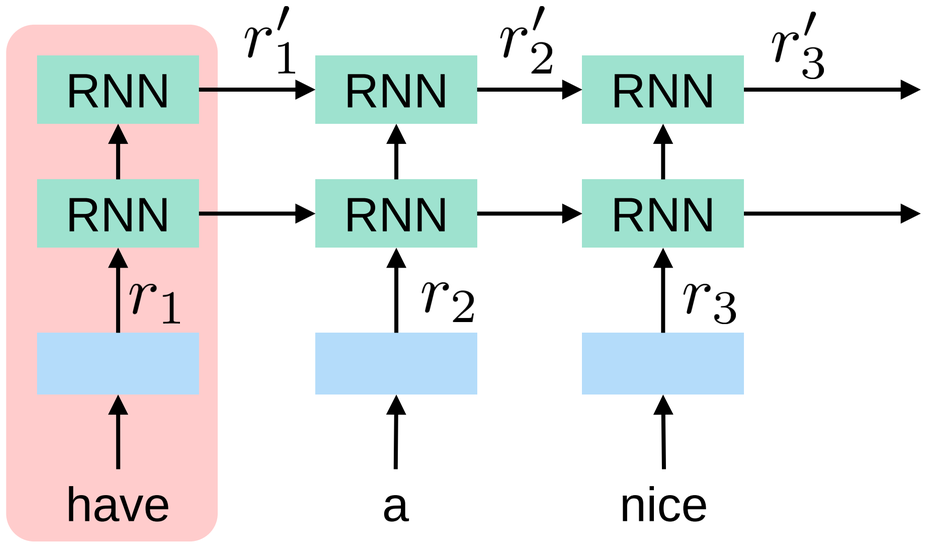
Representations of Variable Length Data

- Input: word sequence, image pixels, audio signal, click logs
- Property: continuity, temporal, importance distribution
- Example
  - ✓ Basic combination: average, sum
  - ✓ Neural combination: network architectures should consider input domain properties
    - CNN (convolutional neural network)
    - RNN (recurrent neural network): temporal information

Network architectures should consider the input domain properties

Recurrent Neural Networks

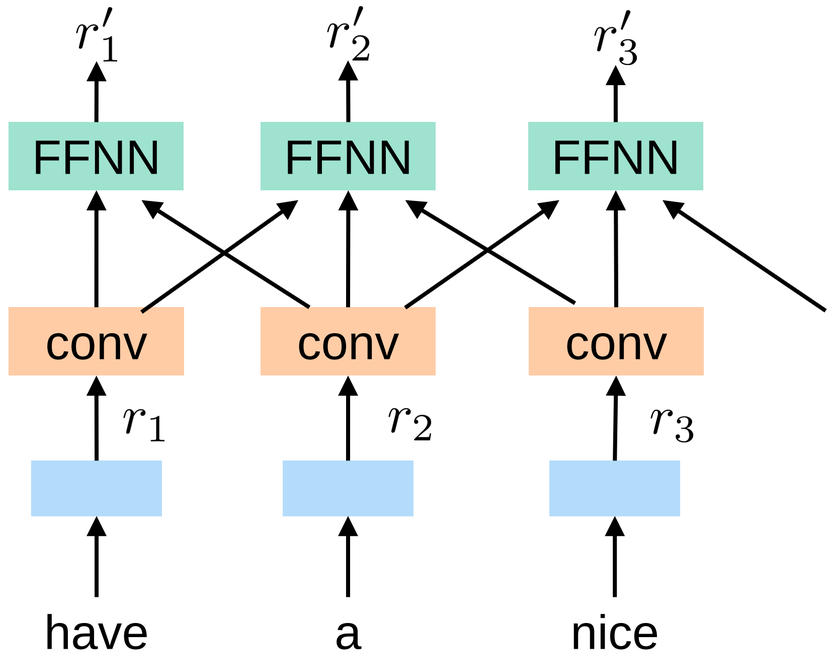
- Learning variable-length representations
  - ✓ Fit for sentences and sequences of values
- Sequential computation makes parallelization difficult
- No explicit modeling of long and short range dependencies



Convolutional Neural Networks

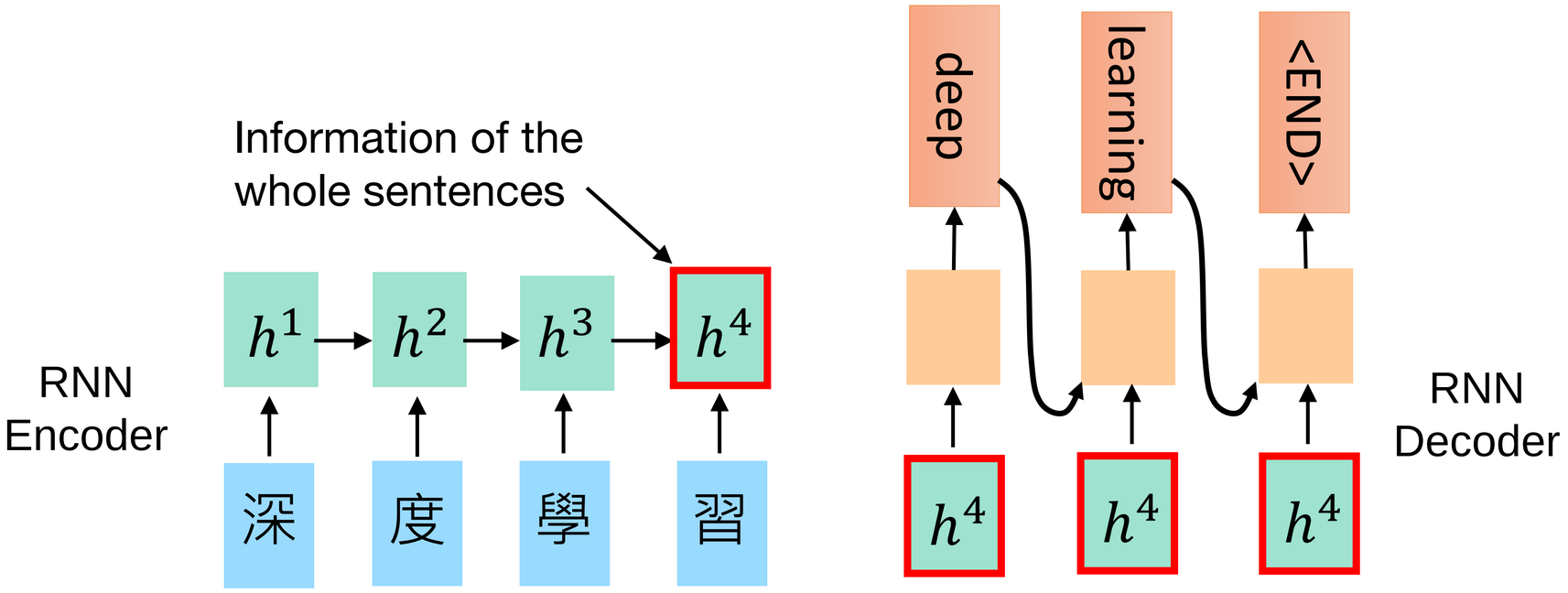
- Easy to parallelize
- Exploit local dependencies

✓ **Long-distance** dependencies require many layers

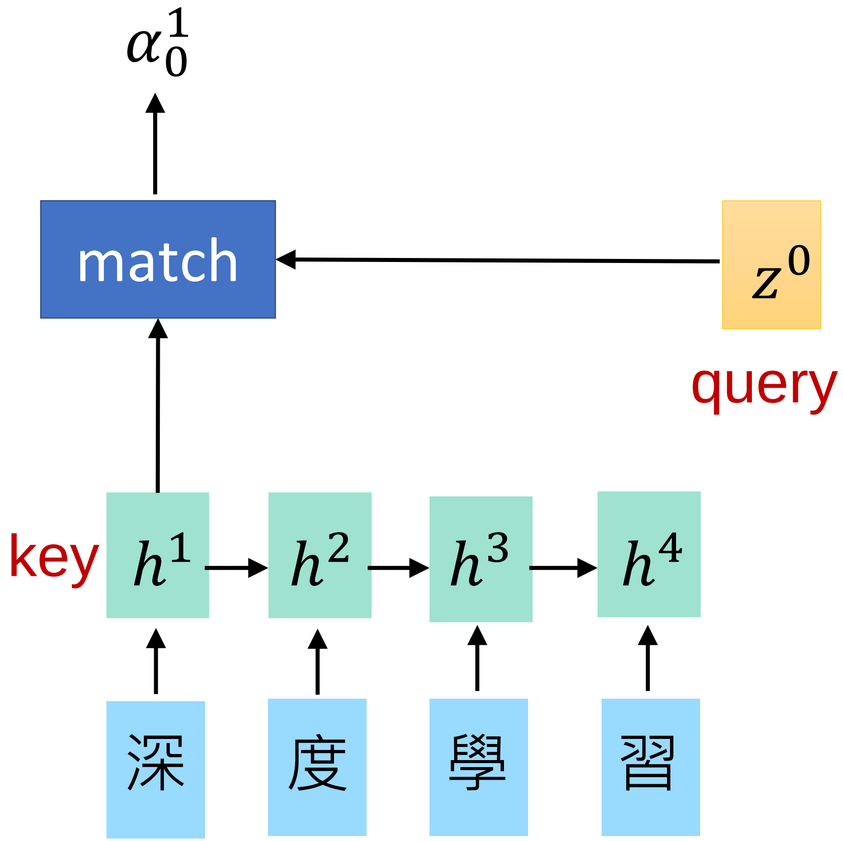


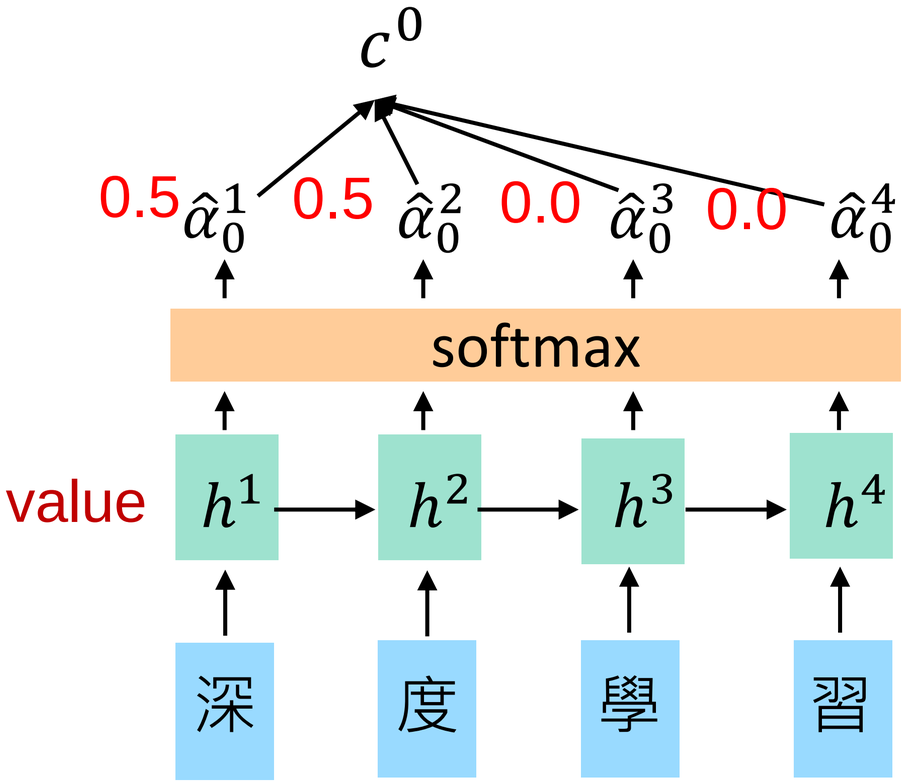
Attention

- Encoder-decoder model is important in NMT
- RNNs need **attention mechanism** to handle long dependencies
- Attention allows us to access any state



Machine Translation with Attention





Dot-Product Attention

- Input: a query q and a set of key-value (k-v) pairs to an output
- Output: weighted sum of values

Inner product of
query and corresponding key

$$A(q, K, V) = \sum_i \frac{\exp(q \cdot k_i)}{\sum_j \exp(q \cdot k_j)} v_i$$

✓ Query q is a d_k-dim vector

✓ Key k is a d_k-dim vector

✓ Value v is a d_v-dim vector

Dot-Product Attention in Matrix

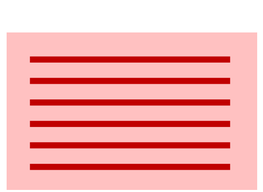
- Input: *multiple* queries q and a set of key-value (k-v) pairs to an output
- Output: a set of weighted sum of values

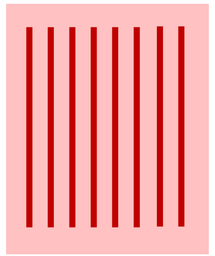
$$A(q, K, V) = \sum_i \frac{\exp(q \cdot k_i)}{\sum_j \exp(q \cdot k_j)} v_i$$

$$A(Q, K, V) = \text{softmax}(QK^T)V$$

$$[|Q| \times d_k] \times [d_k \times |K|] \times [|K| \times d_v]$$

softmax
row-wise





$$= [|Q| \times d_v]$$