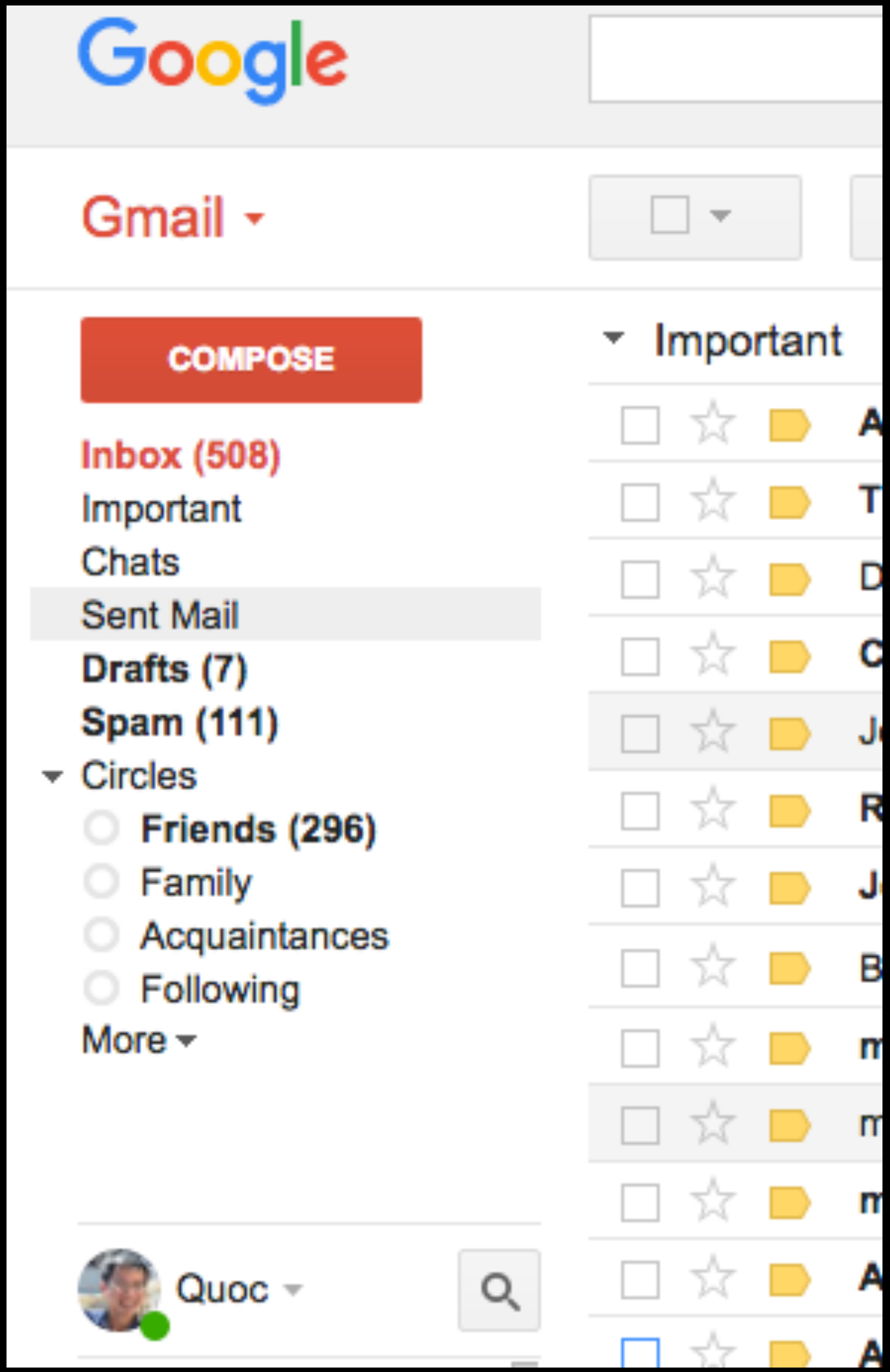
Sequence to Sequence Learning for NLP and Speech

Quoc V. Le
Google Brain team

“AutoReply”



- 508 unread emails!!!
- Some emails just require “Yes” / “No” answers
- Let’s build “AutoReply”

“AutoReply”

- From: Ann
- Subject: Hi
- Content: Are you visiting Vietnam for the new year, Quoc?
- Probable Reply: Yes

Dataset

- Are you visiting Vietnam for the new year, Quoc? -> Yes
- Are you hanging out with us tonight? -> No
- Did you read the cool paper on ResNet? -> Yes
- …

Preprocessing

- Are you visiting Vietnam for the new year , Quoc ? -> Yes
- Are you hanging out with us tonight ? -> No
- Did you read the cool paper on ResNet ? -> Yes
- …

Preprocessing

- Are you visiting Vietnam for the new year , Quoc ? -> Yes
- Are you hanging out with us tonight ? -> No
- Did you read the cool paper on ResNet ? -> Yes
- …

Feature Representation

Are you visiting Vietnam for the new year , Quoc ?

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 0, 0, 1, 0, 0, 0, 2]

20,000 dimensions

Feature Representation

Are you visiting Vietnam for the new year , Quoc ?

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 0, 0, 1, 0, 0, 0, 2]

20,000 dimensions

Feature Representation

Are you visiting Vietnam for the new year , Quoc ?

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 0, 0, 1, 0, 0, 0, 2]

20,000 dimensions

Feature Representation

Are you visiting Vietnam for the new year , Quoc ?

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 0, 0, 1, 0, 0, 0, 2]

20,000 dimensions

Feature Representation

Are you visiting Vietnam for the new year , Quoc ?

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ... , 0, 0, 1, 0, 0, 0, 2]

20,000 dimensions

Special dimension reserved for out of vocabulary words

Formulation

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 0, 0, 1, 0, 0, 0, 2] -> 1

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 1, 0, 0, 0, 0, 0, 0] -> 0

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, …. , 0, 3, 0, 0, 0, 0, 1] -> 1

Formulation

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 0, 0, 1, 0, 0, 0, 2] -> 1

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, …. , 1, 0, 0, 0, 0, 0, 0] -> 0

[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, …. , 0, 3, 0, 0, 0, 0, 1] -> 1

x y

Formulation

- Find W such that Wx approximates y
- Since y is in {"Yes", "No"}, this is a "Logistic Regression" problem

$$\begin{bmatrix} \dfrac{\exp(w_1^T x)}{\exp(w_1^T x) + \exp(w_2^T x)} \\ \dfrac{\exp(w_2^T x)}{\exp(w_1^T x) + \exp(w_2^T x)} \end{bmatrix}$$

Formulation

- Find W such that Wx approximates y
- Since y is in {"Yes", "No"}, this is a "Logistic Regression" problem

$$\begin{bmatrix} \dfrac{\exp(w_1^T x)}{\exp(w_1^T x) + \exp(w_2^T x)} \\ \dfrac{\exp(w_2^T x)}{\exp(w_1^T x) + \exp(w_2^T x)} \end{bmatrix}$$

Positive and sum up to 1

Training with stochastic gradient descent

- For iteration 1, 2, 3, …, 1000000
  - Sample a random email x and a reply
  - If reply == Yes, update w_1 and w_2 to increase $$\frac{\exp(w_1^T x)}{\exp(w_1^T x) + \exp(w_2^T x)}$$
  - If reply == No, update w_1 and w_2 to increase $$\frac{\exp(w_2^T x)}{\exp(w_1^T x) + \exp(w_2^T x)}$$

Training with stochastic gradient descent

- For iteration 1, 2, 3, …, 1000000
  - Sample a random email x and a reply
  - If reply == Yes, update w_1 and w_2 to increase

$$p_1 = \frac{\exp(w_1^T x)}{\exp(w_1^T x) + \exp(w_2^T x)}$$

  - If reply == No, update w_1 and w_2 to increase

$$p_2 = \frac{\exp(w_2^T x)}{\exp(w_1^T x) + \exp(w_2^T x)}$$

Training with stochastic gradient descent

- For iteration 1, 2, 3, …, 1000000
  - Sample a random email x and a reply
  - If reply == Yes, update w_1 and w_2

$$w_1 = w_1 + \text{alpha} \frac{d \log(p_1)}{d\, w_1} \qquad w_2 = w_2 + \text{alpha} \frac{d \log(p_1)}{d\, w_2}$$

  - If reply == No, update w1 and w2

$$w_1 = w_1 + \text{alpha} \frac{d \log(p_2)}{d\, w_1} \qquad w_2 = w_2 + \text{alpha} \frac{d \log(p_2)}{d\, w_2}$$

Prediction

- For any incoming email x
  - Compute $$\frac{\exp(w_1^T x)}{\exp(w_1^T x) + \exp(w_2^T x)}$$
  - If > 0.5 -> reply = Yes
  - If <= 0.5 -> reply = No

Information Loss

Are you visiting Vietnam for the new year , Quoc ?

[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ... , 0, 0, 1, 0, 0, 0, 2]

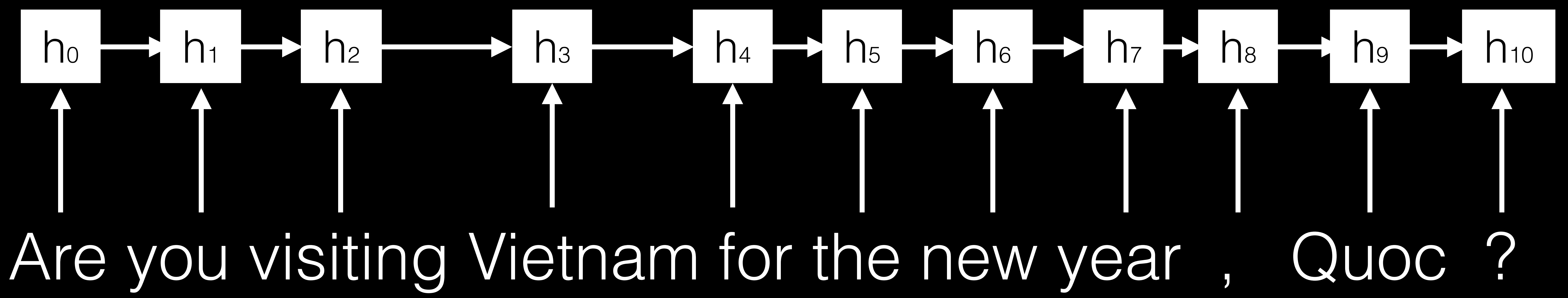
20,000 dimensions

This "bag-of-words representation" does not care about the order of the words!

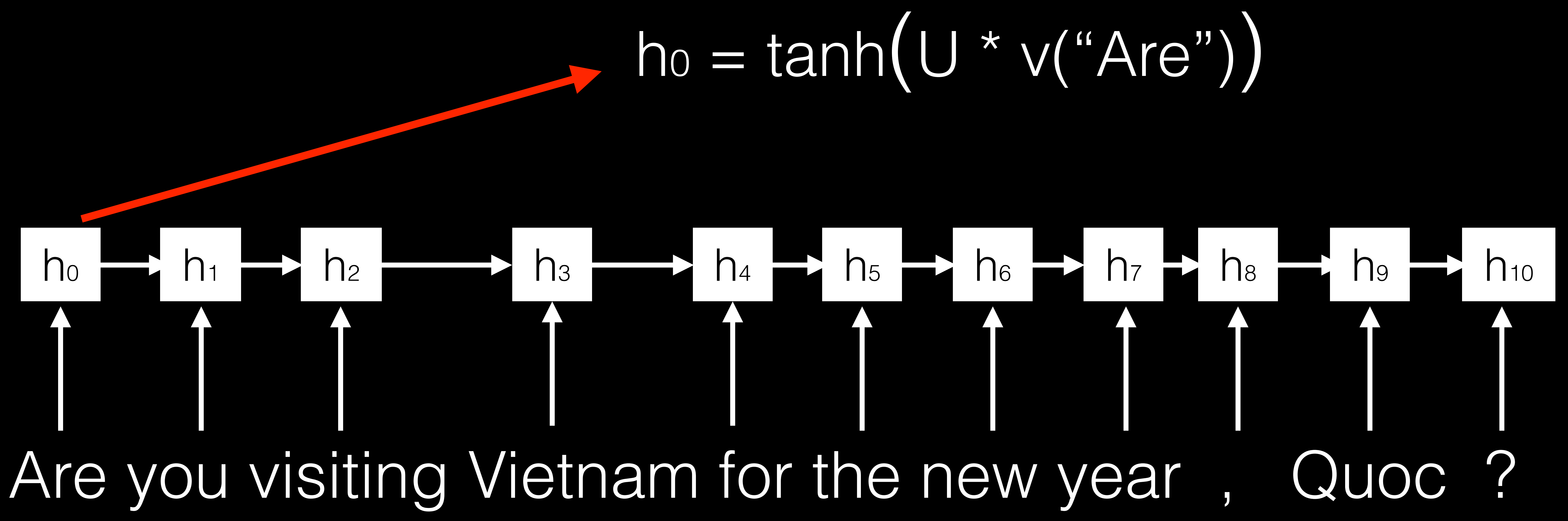
Recurrent Neural Network

Are you visiting Vietnam for the new year , Quoc ?

Recurrent Neural Network

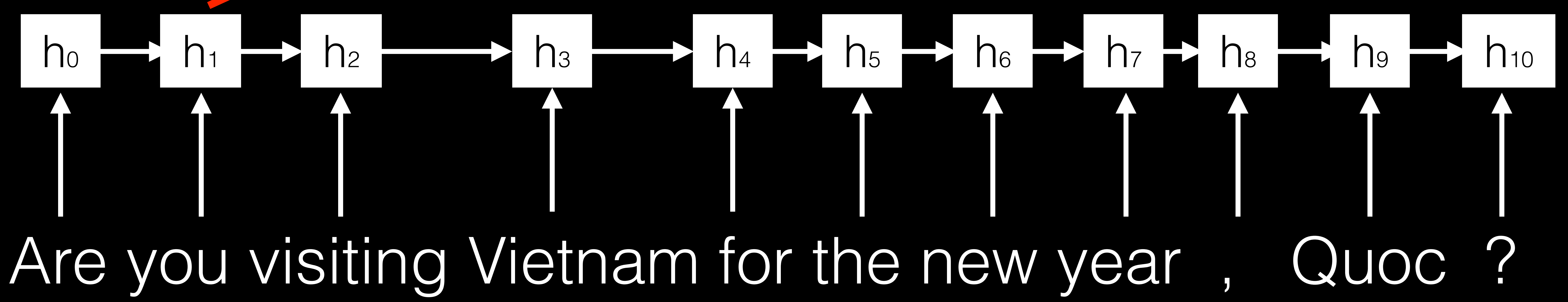


Recurrent Neural Network



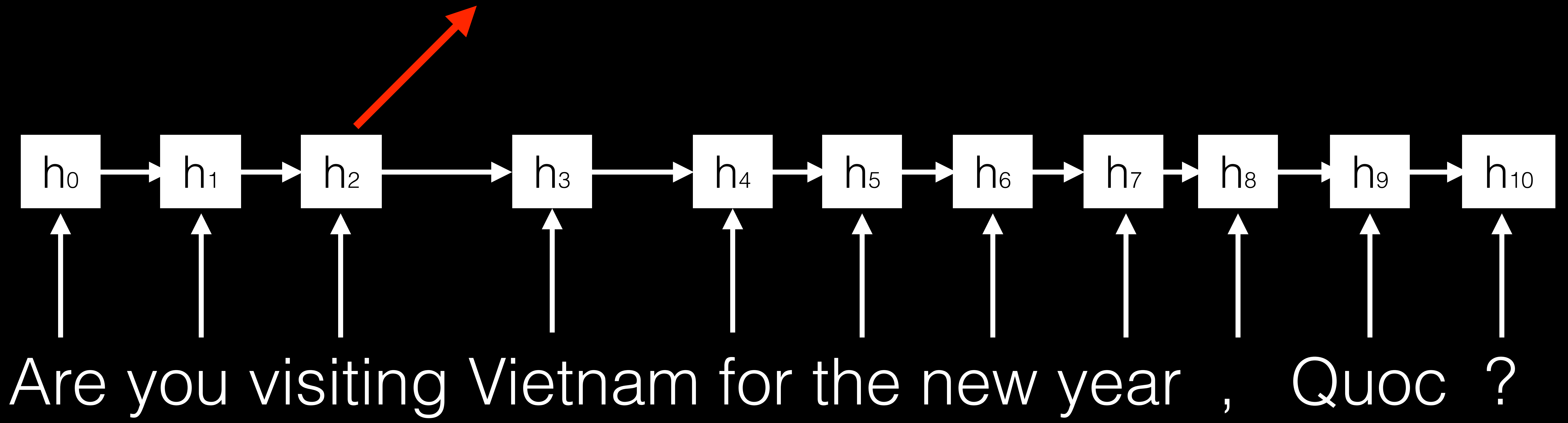
Recurrent Neural Network

$$h_1 = \tanh\left(A^* \, h_0 + U * v(\text{“you”})\right)$$

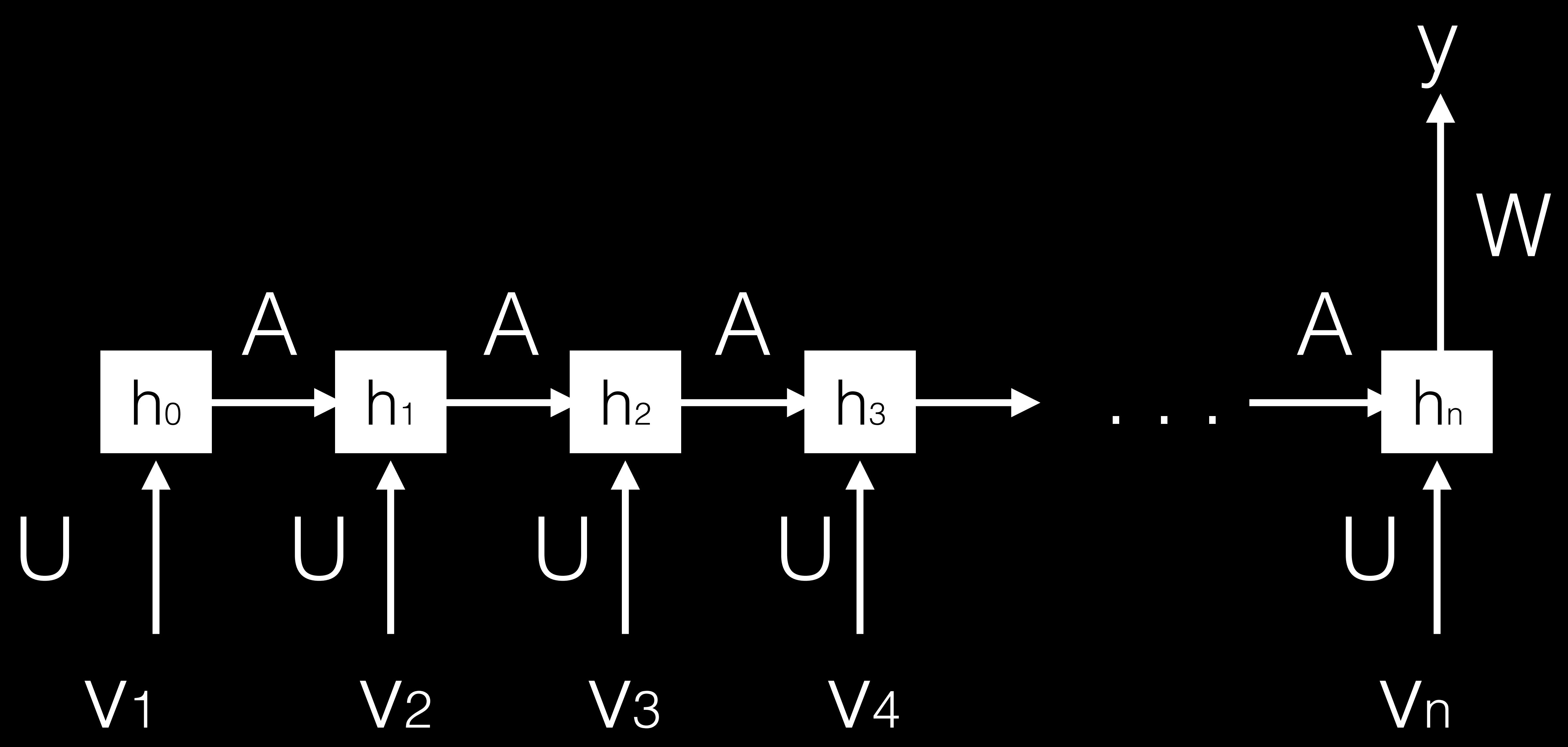


Recurrent Neural Network

$$h_2 = \tanh\left(A^* \, h_1 + U * v(\text{“visting”})\right)$$



Recurrent Neural Network



Training RNN with stochastic gradient descent

- For iteration 1, 2, 3, …, 1000000
  - Sample a random email x and a reply
  - If reply == Yes, update w_1 and w_2

$$w_1 = w_1 + \text{alpha} \frac{d \log(p_1)}{d\, w_1} \qquad w_2 = w_2 + \text{alpha} \frac{d \log(p_1)}{d\, w_2}$$

Training RNN with stochastic gradient descent

- For iteration 1, 2, 3, …, 1000000
  - Sample a random email x and a reply
  - If reply == Yes, update w_1 and w_2

$$w_1 = w_1 + \text{alpha} \frac{d \log(p_1)}{d\, w_1} \qquad w_2 = w_2 + \text{alpha} \frac{d \log(p_1)}{d\, w_2}$$

Update U, and A

$$A = A + \text{alpha} \frac{d \log(p_1)}{d\, A} \qquad U = U + \text{alpha} \frac{d \log(p_1)}{d\, U}$$

Update all relevant v's

$$v_i = v_i + \text{alpha} \frac{d \log(p_1)}{d\, v_i}$$

Training RNN with stochastic gradient descent

- For iteration 1, 2, 3, …, 1000000
  - Sample a random email x and a reply
  - If reply == Yes, update w_1 and w_2

$$w_1 = w_1 + \text{alpha} \frac{d \log(p_1)}{d\, w_1} \qquad w_2 = w_2 + \text{alpha} \frac{d \log(p_1)}{d\, w_2}$$

Update U, and A

$$A = A + \text{alpha} \frac{d \log(p_1)}{d\, A} \qquad U = U + \text{alpha} \frac{d \log(p_1)}{d\, U}$$

Update all relevant v's

$$v_i = v_i + \text{alpha} \frac{d \log(p_1)}{d\, v_i}$$

Very hard to derive! Use autodiff :)

The big picture so far

- Bag-of-word representation
- RNN representation for variable-sized input
- Autodiff to compute the partial derivatives (TensorFlow, Theano, Torch)
- Stochastic gradient descent for training

More friendly “AutoReply”

- Are you visiting Vietnam for the new year , Quoc ? -> Yes , see you soon !
- Are you hanging out with us tonight ? -> No , I am too busy .
- Did you read the cool paper on ResNet? -> Yes , it’s nice !
- …

Better Formulation

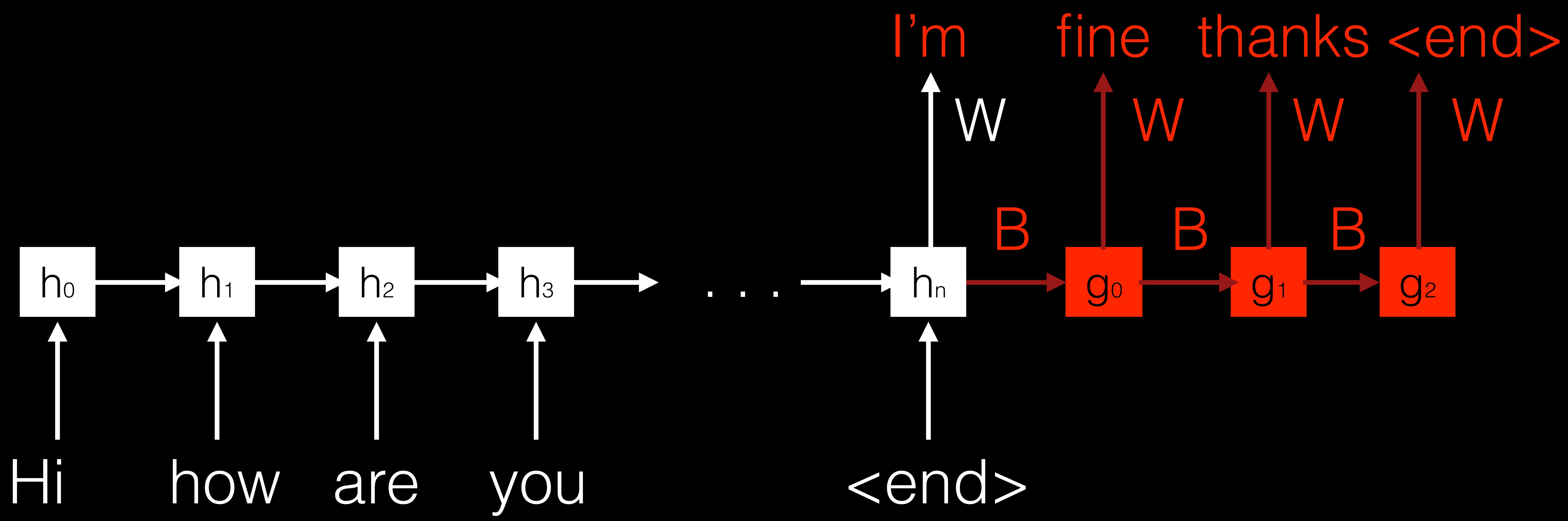
- Mapping between variable-length input to variable length output

Better Formulation

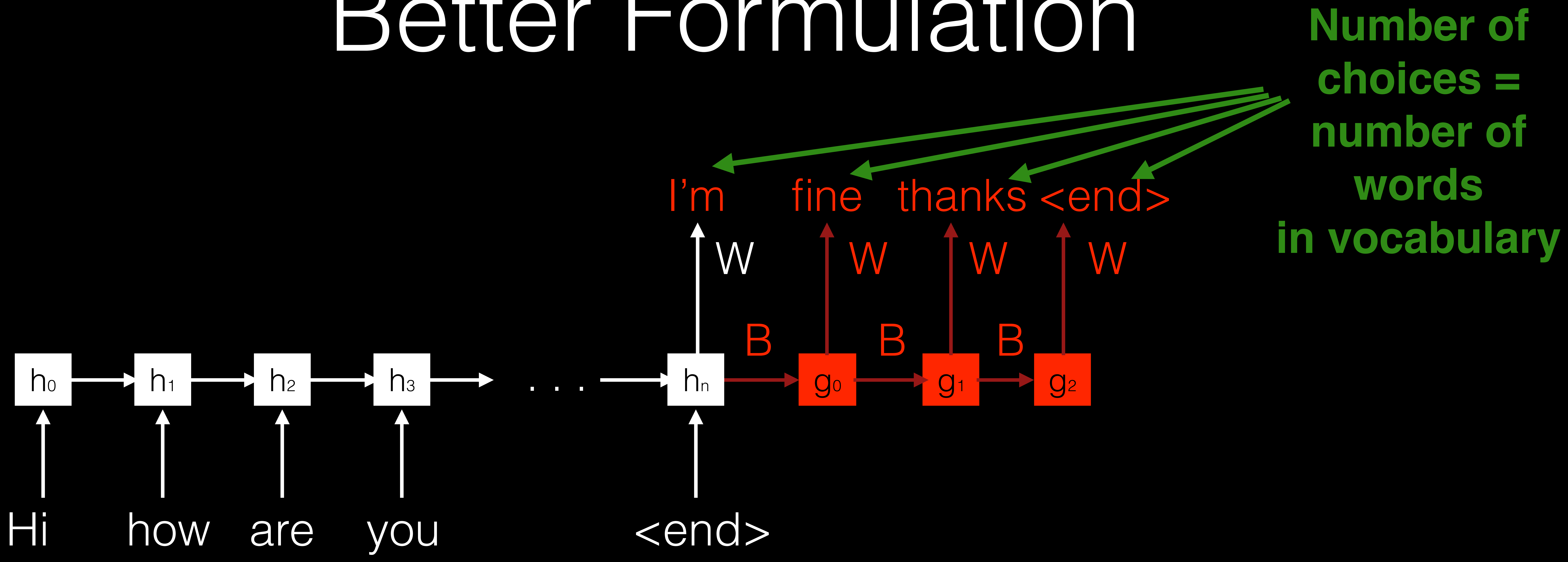
- Mapping between variable-length input to variable length output
- Applications: AutoReply, Translation, Image Captioning, Summarization, Speech Transcription, Conversation, Q&A, …

Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks

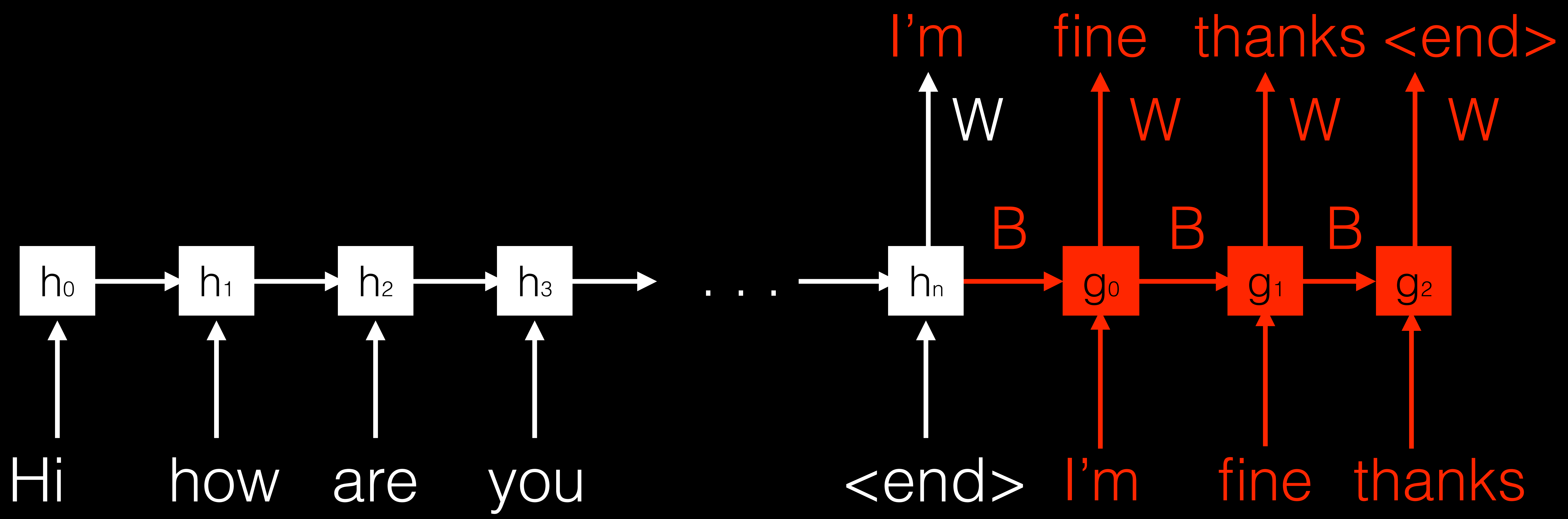
Better Formulation



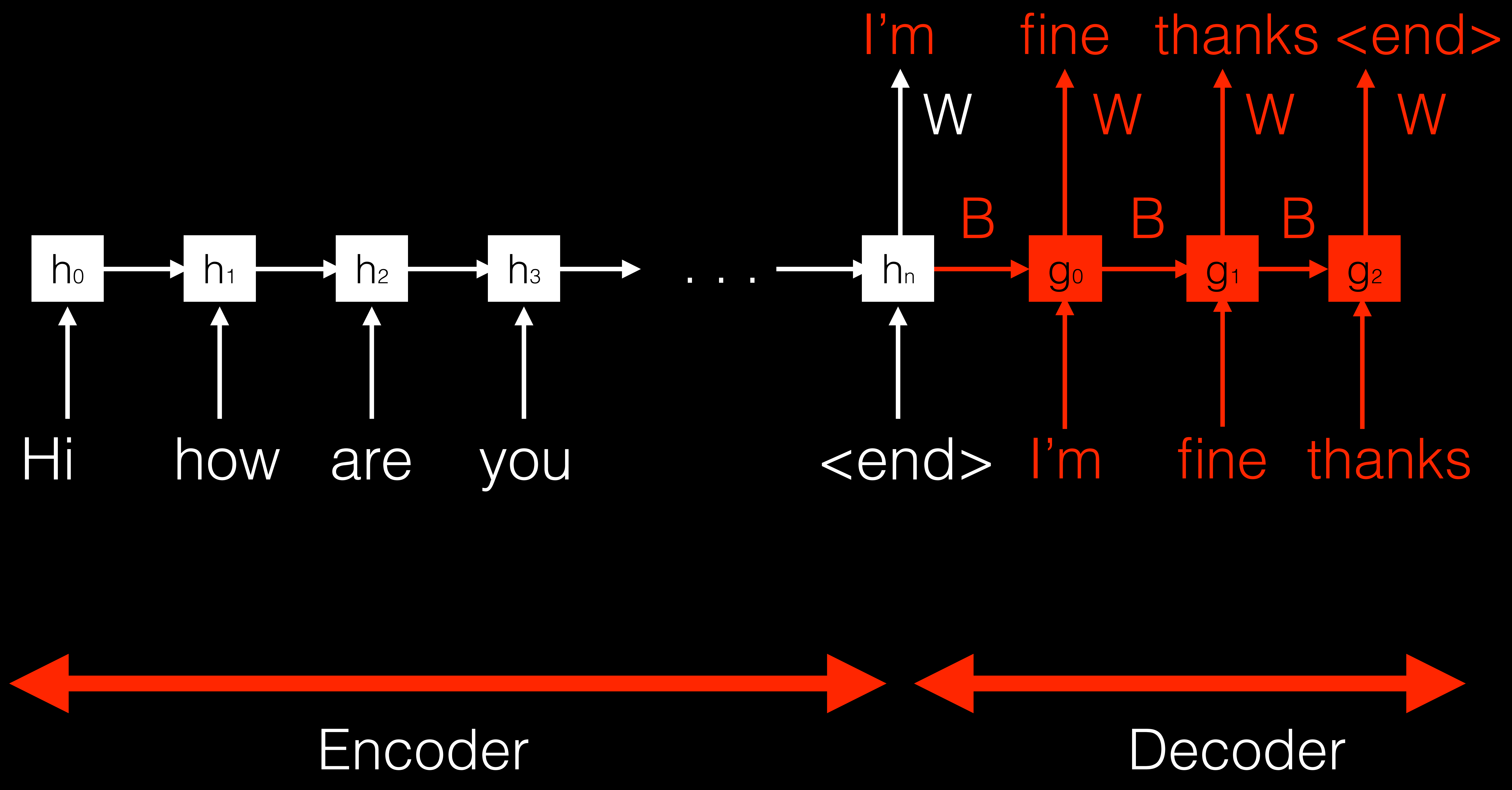
Better Formulation



Better Formulation



Better Formulation



Sequence to Sequence Training with SGD

- For iteration 1, 2, 3, …., 1000000
  - Sample an email **x** and a reply **y**
  - Sample a random word $\mathbf{y}_{(t)}$ in **y**
  - Update RNN encoder and decoder parameters to increase the probability of word $\mathbf{y}_{(t)}$ given $\mathbf{y}_{(t-1)}, \mathbf{y}_{(t-2)}, \ldots, \mathbf{y}_{(0)}, \mathbf{x}_{(n)}, \mathbf{x}_{(n-1)}, \ldots, \mathbf{x}_{(0)}$ using partial derivative with respect to **W, U, A, B,** and all **v**’s

Sequence to Sequence Training with SGD

- For iteration 1, 2, 3, …., 1000000
  - Sample an email **x** and a reply **y**
  - Sample a random word $\mathbf{y}_{(t)}$ in **y**
  - Update RNN encoder and decoder parameters to increase the probability of word $\mathbf{y}_{(t)}$ given $\mathbf{y}_{(t-1)}, \mathbf{y}_{(t-2)}, \ldots, \mathbf{y}_{(0)}, \mathbf{x}_{(n)}, \mathbf{x}_{(n-1)}, \ldots, \mathbf{x}_{(0)}$ using partial derivative with respect to **W, U, A, B,** and all **v**'s

Very hard to derive! Use autodiff :)

Sequence to Sequence Prediction

- For any incoming email **x**
  - Given **x**, find word $\mathbf{y}_{(0)}$ with highest probability using RNN
  - Given $\mathbf{y}_{(0)}$ and **x**, find word $\mathbf{y}_{(1)}$ with highest probability using RNN
  - …
  - Stop when see **<end>**

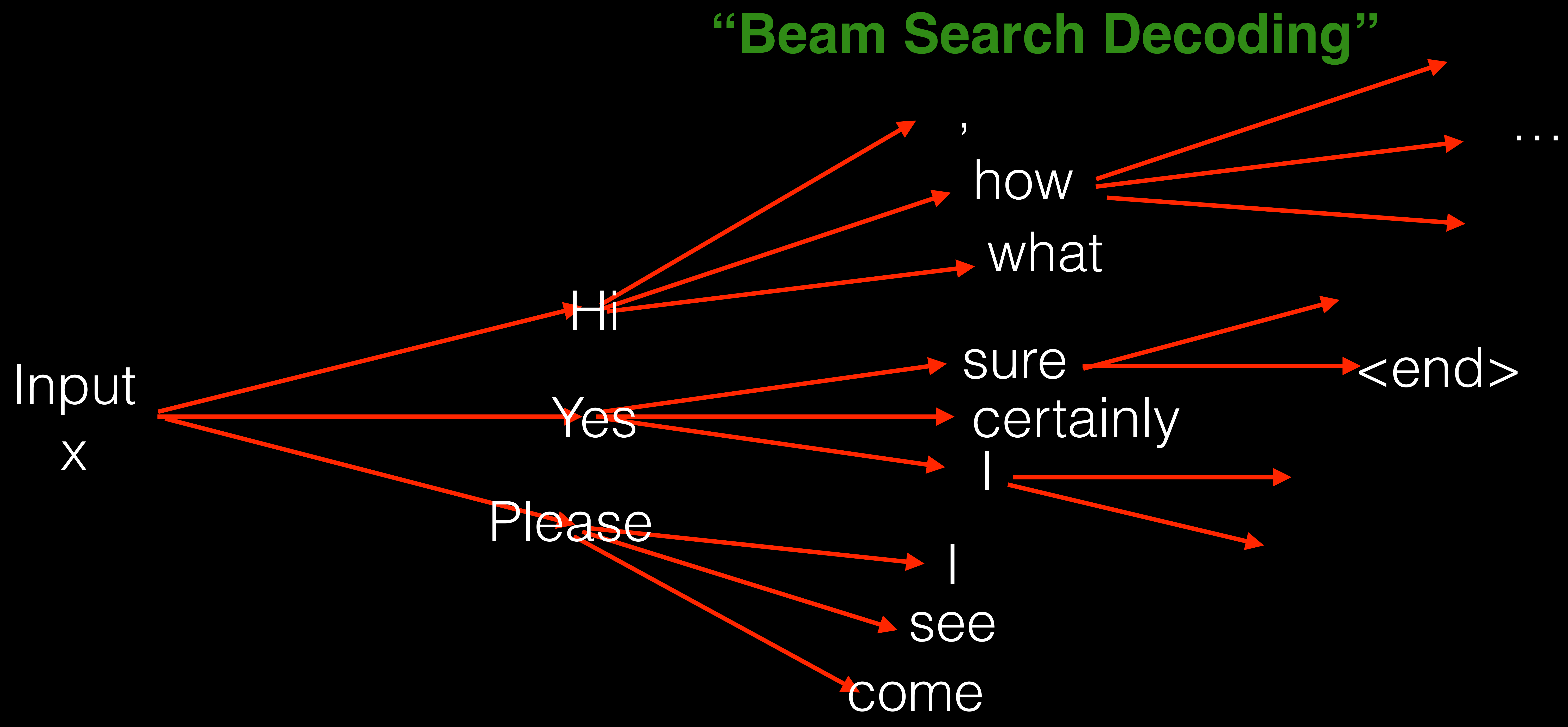
“Greedy Decoding”

Sequence to Sequence Prediction

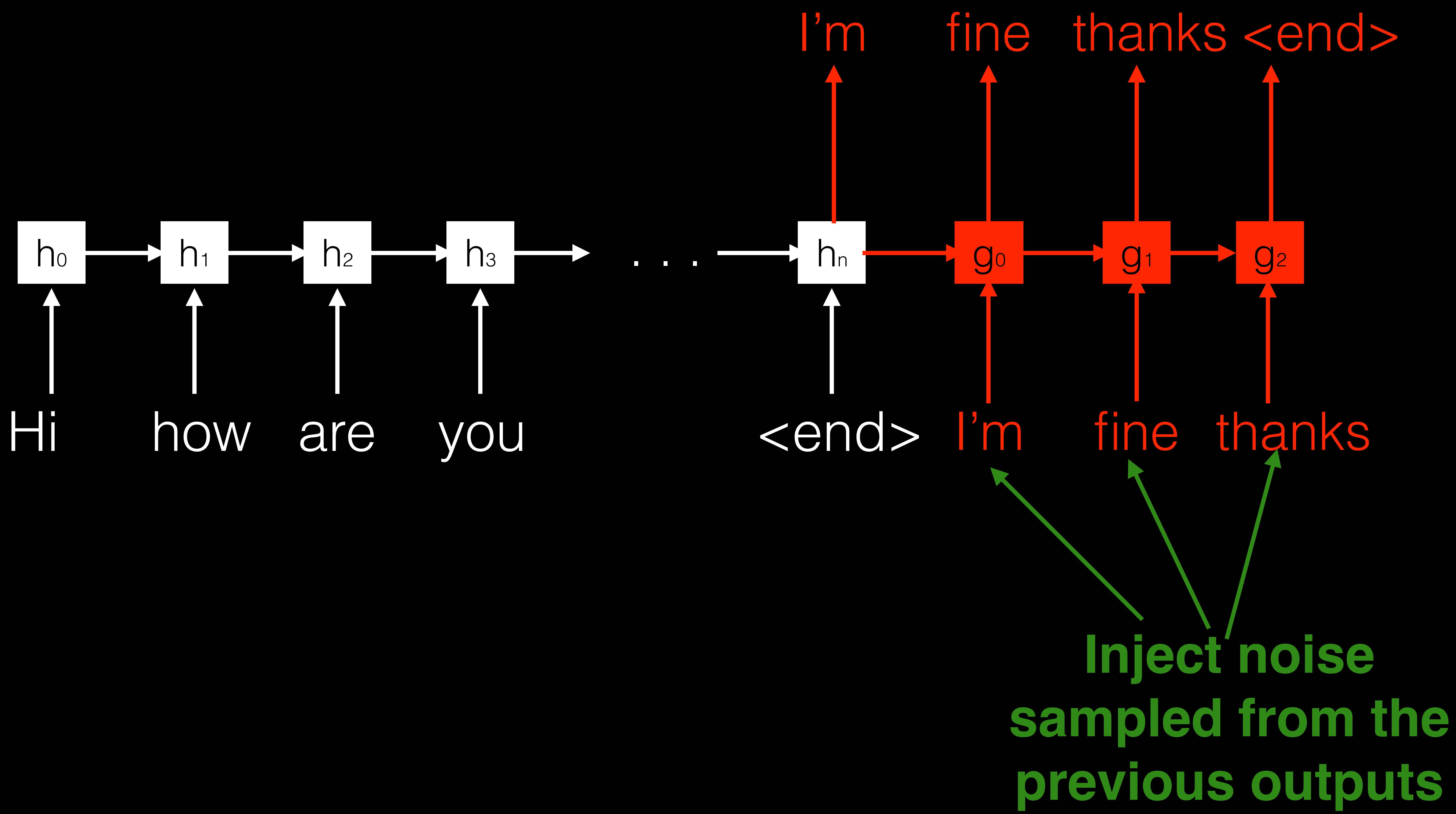
- For any incoming email **x**
  - Given **x**, find **k** candidates for $\mathbf{y}_{(0)}$ with highest probability using RNN
  - Given **x**, for each candidate $\mathbf{y}_{(0)}$, find **k** candidates for word $\mathbf{y}_{(1)}$ with highest probability using RNN
  - …
  - Stop when see **<end>** on each beam
  - Reply = beam with highest probability

“Beam Search Decoding”

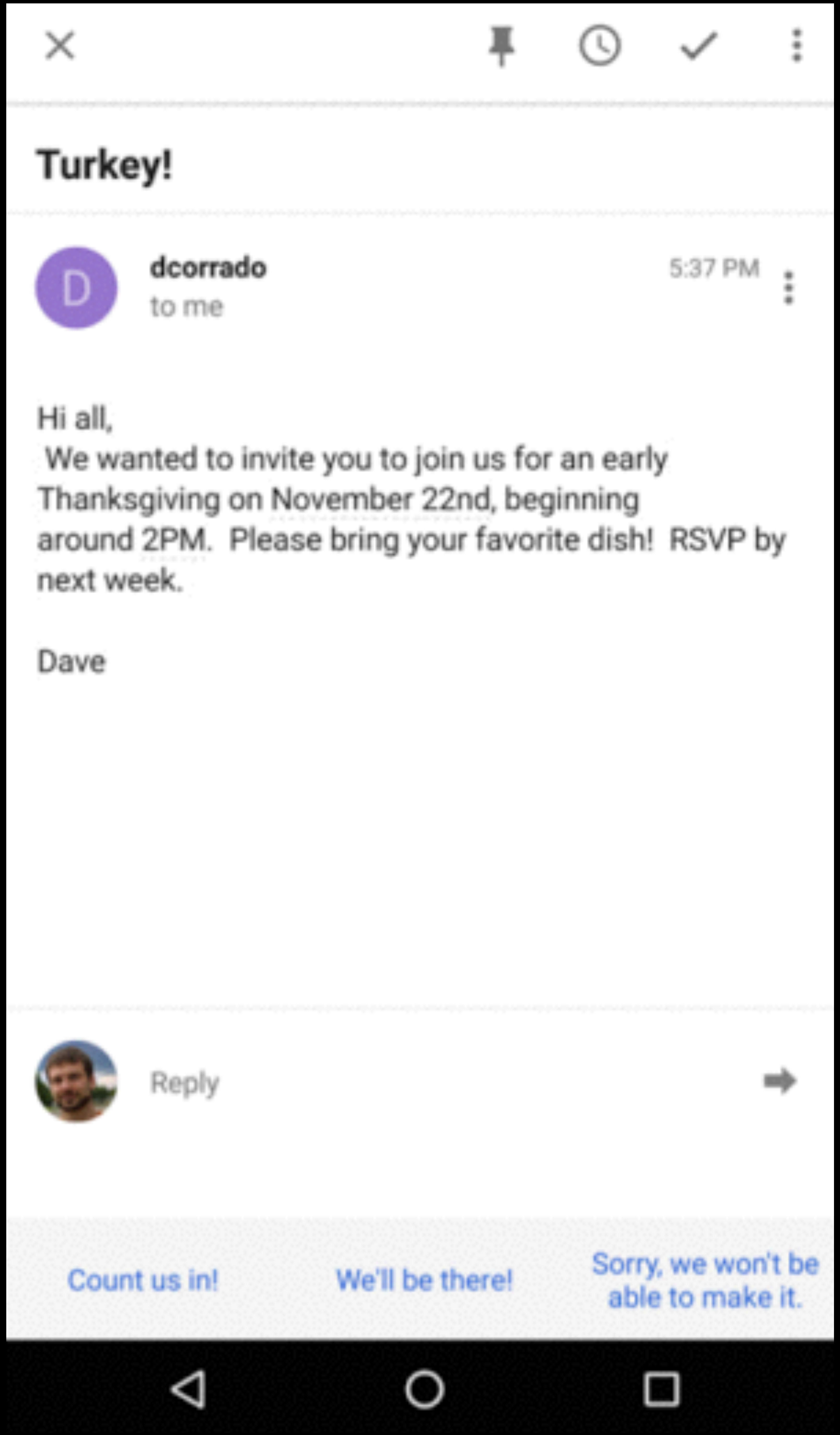
Sequence to Sequence Prediction



Scheduled Sampling



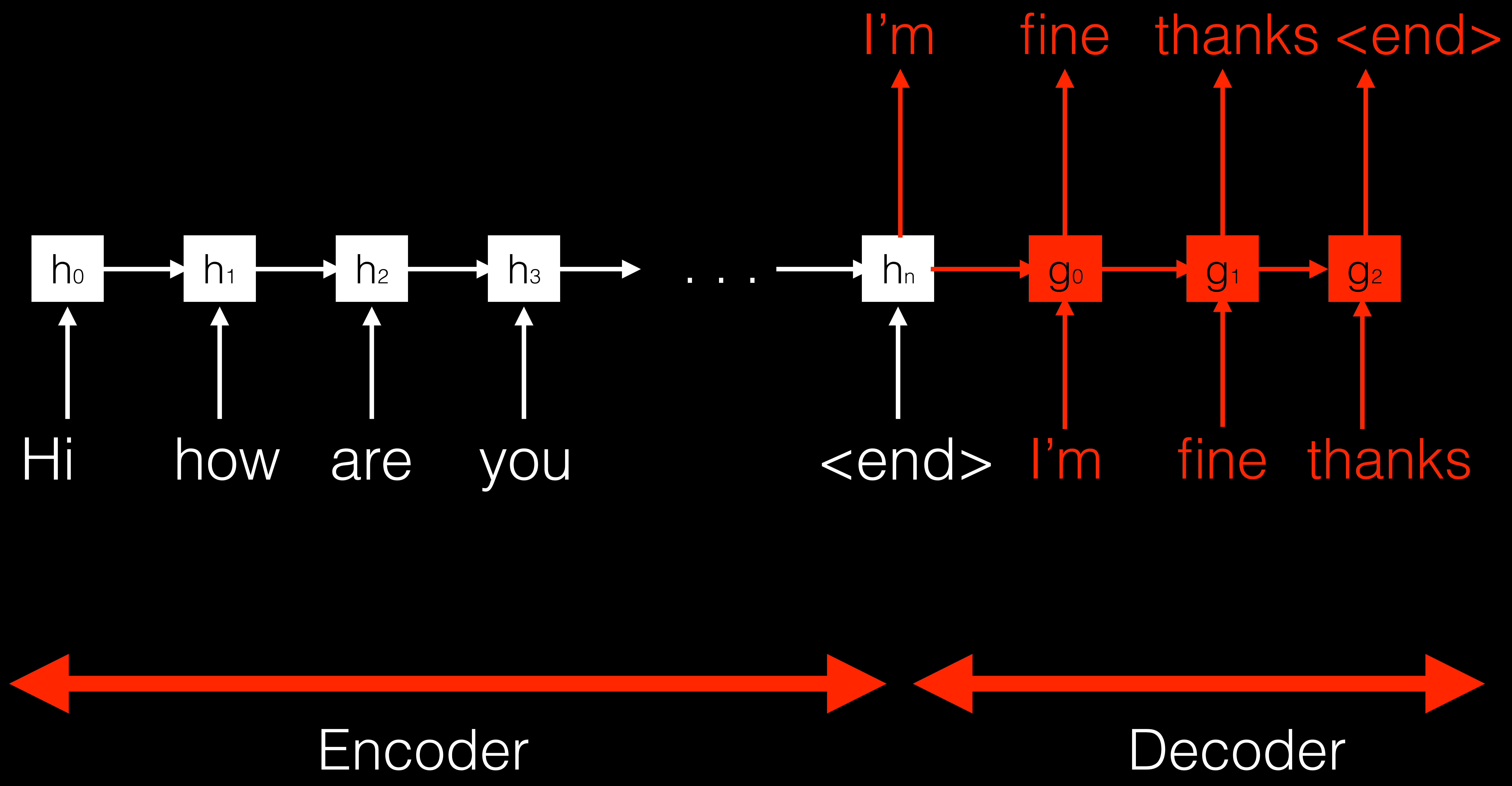
SmartReply feature in Inbox



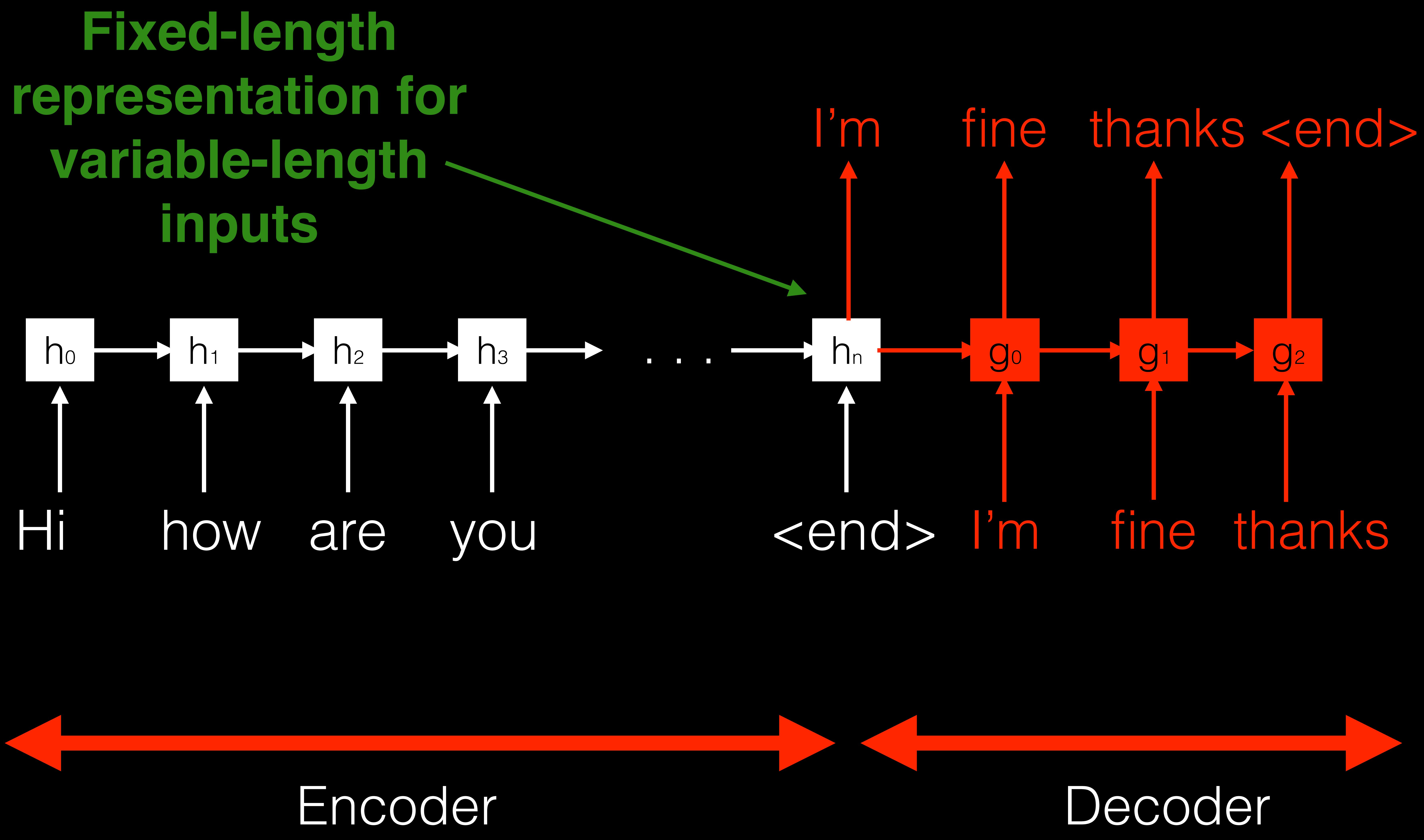
The big picture so far

- RNN encoder and RNN decoder for sequence to sequence learning
- Use stochastic gradient descent for training
- Beam search decoding

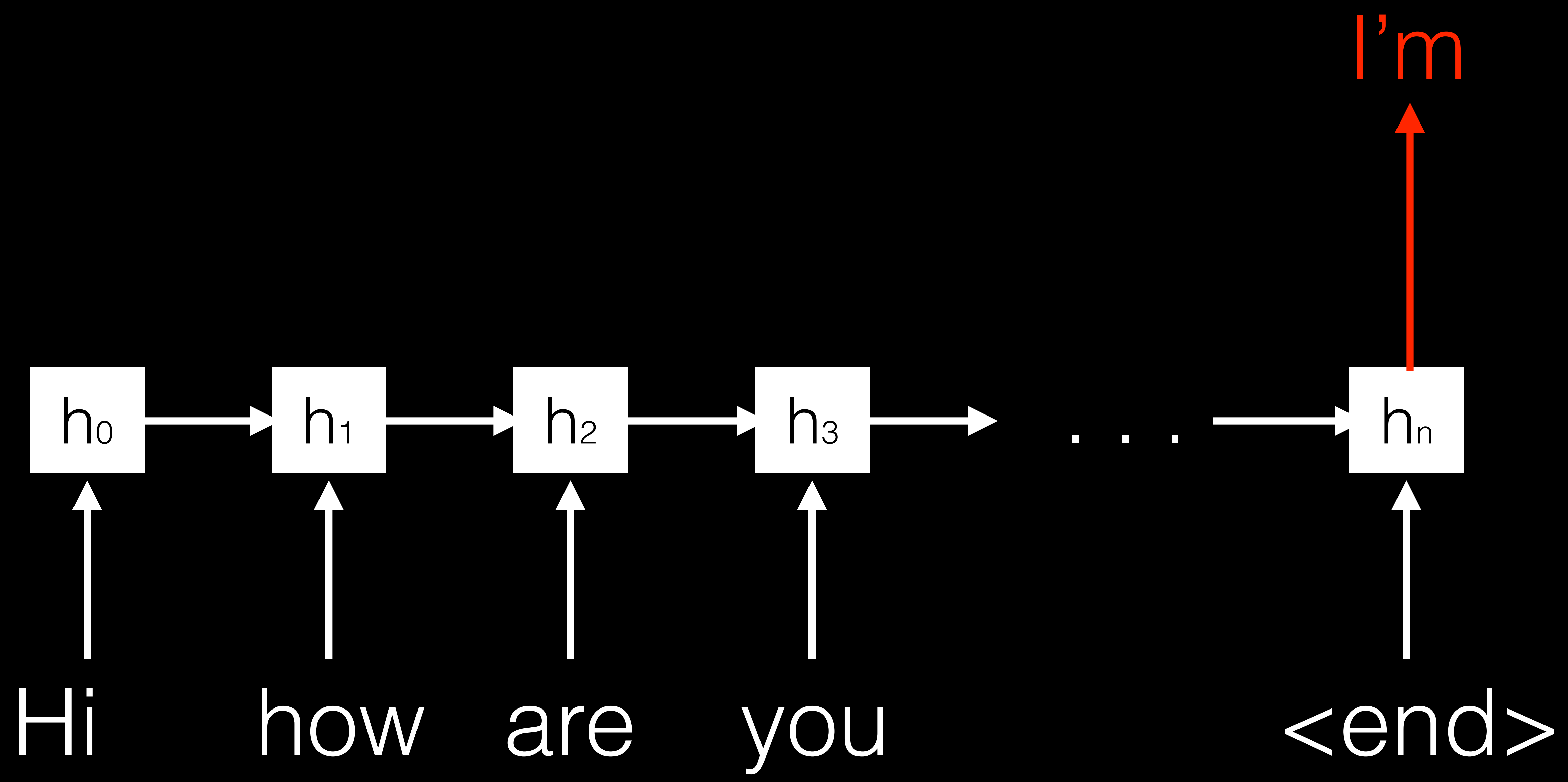
Attention Mechanism



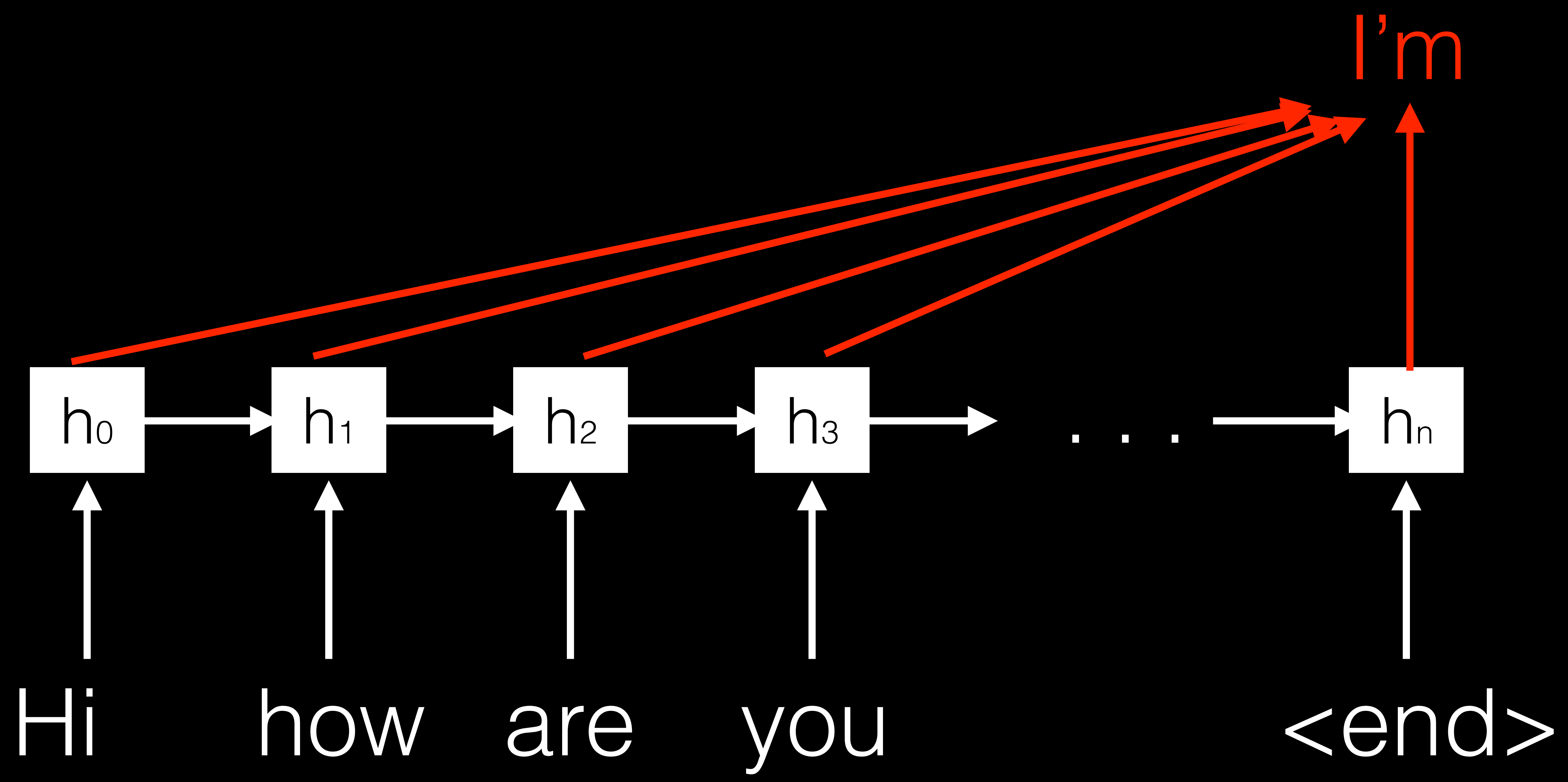
Attention Mechanism



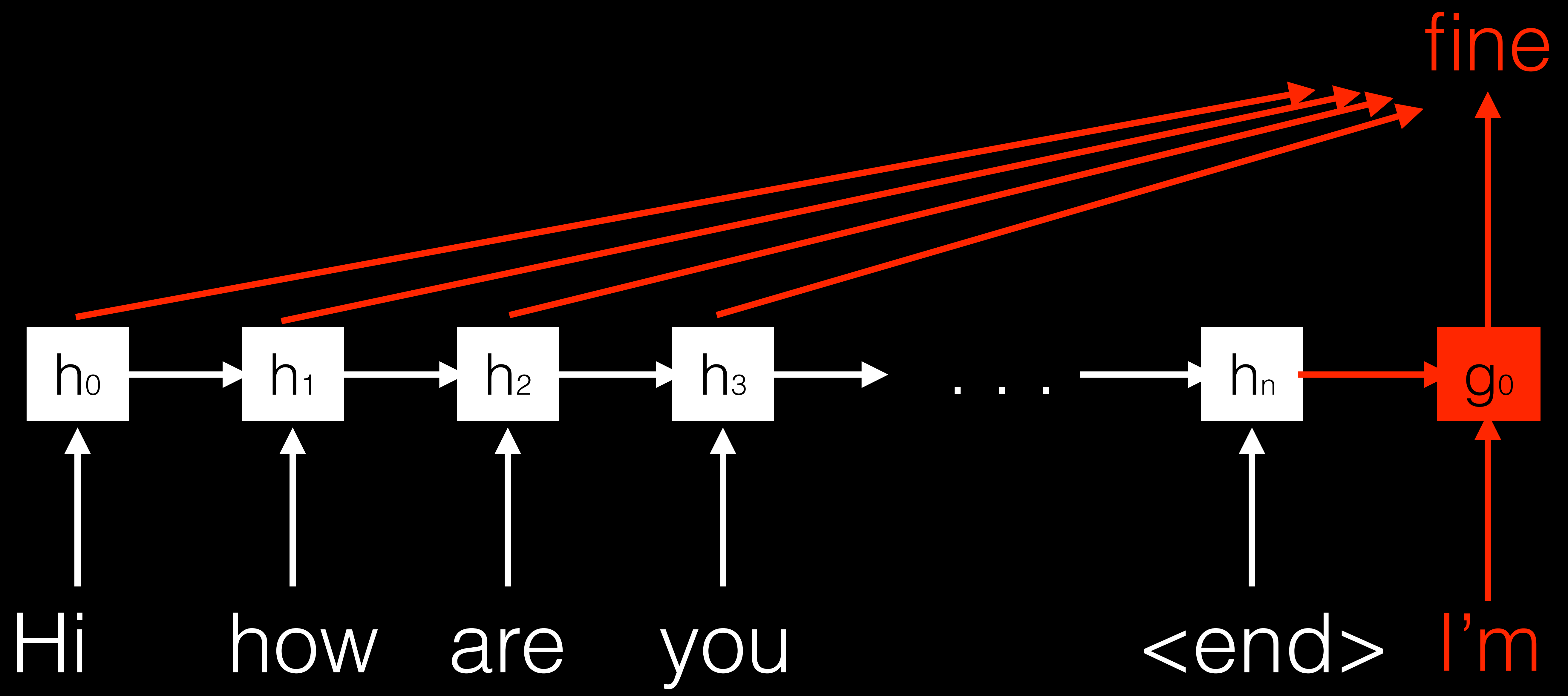
Attention Mechanism



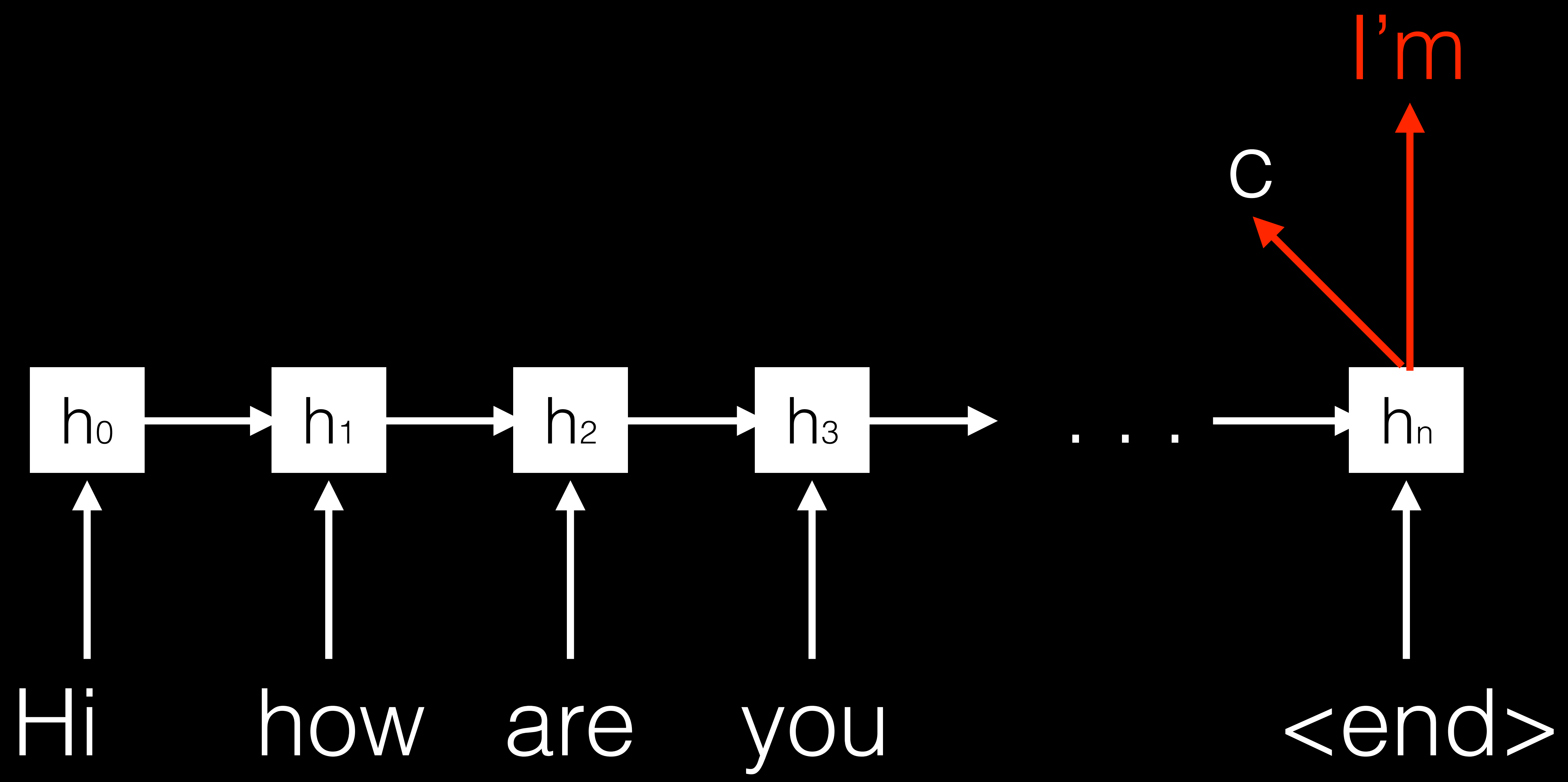
Attention Mechanism



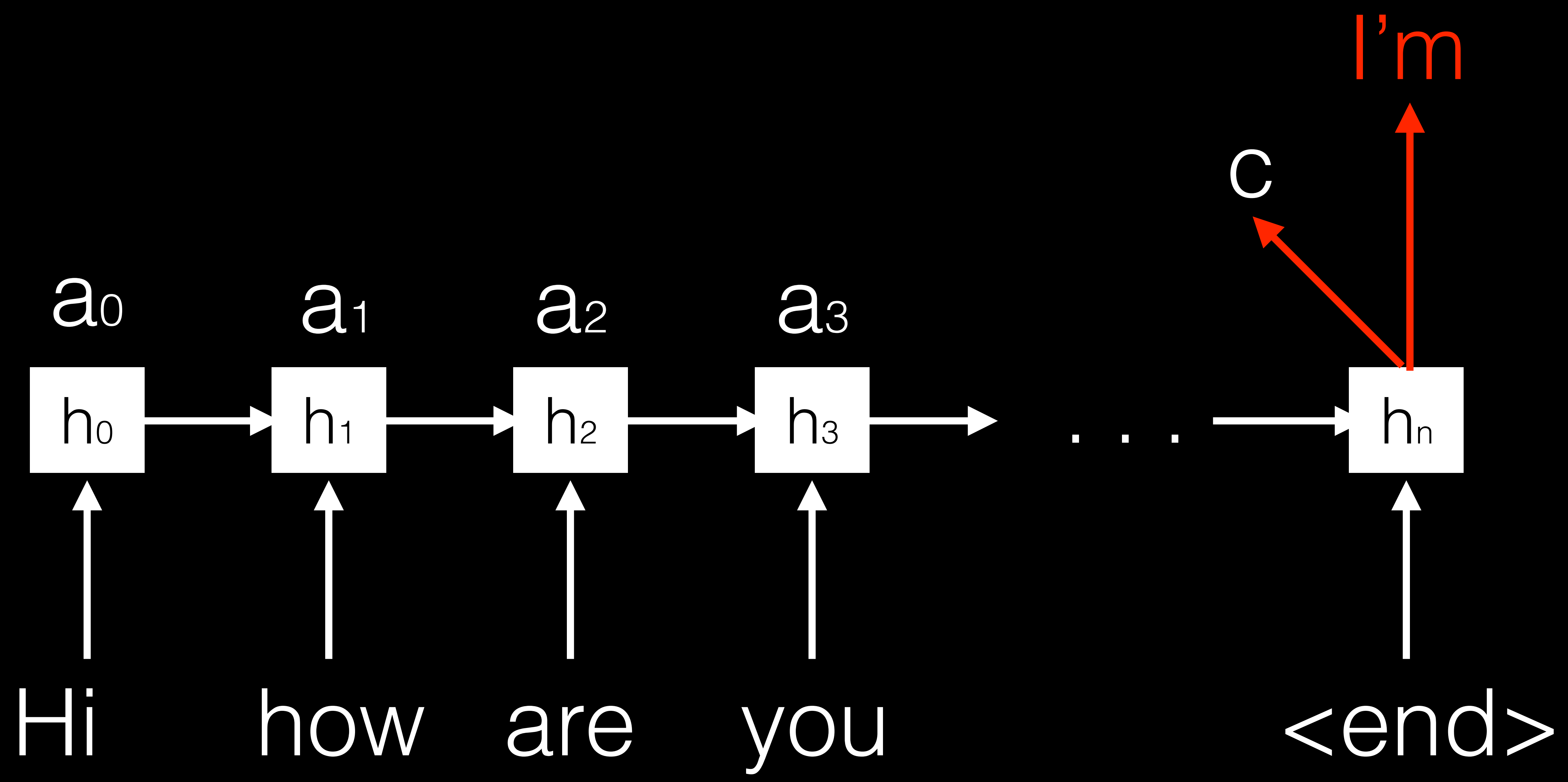
Attention Mechanism



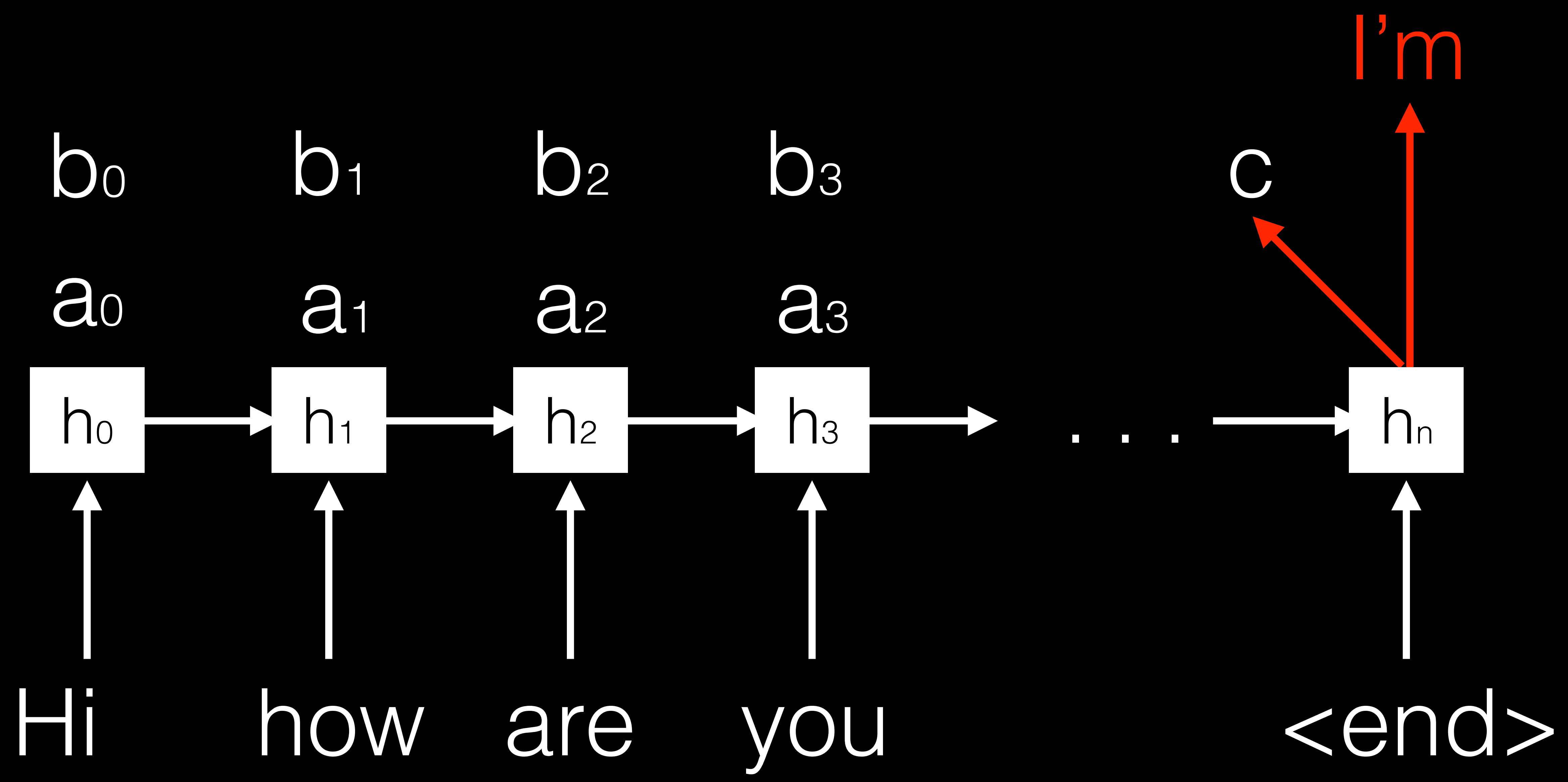
Attention Mechanism



Attention Mechanism

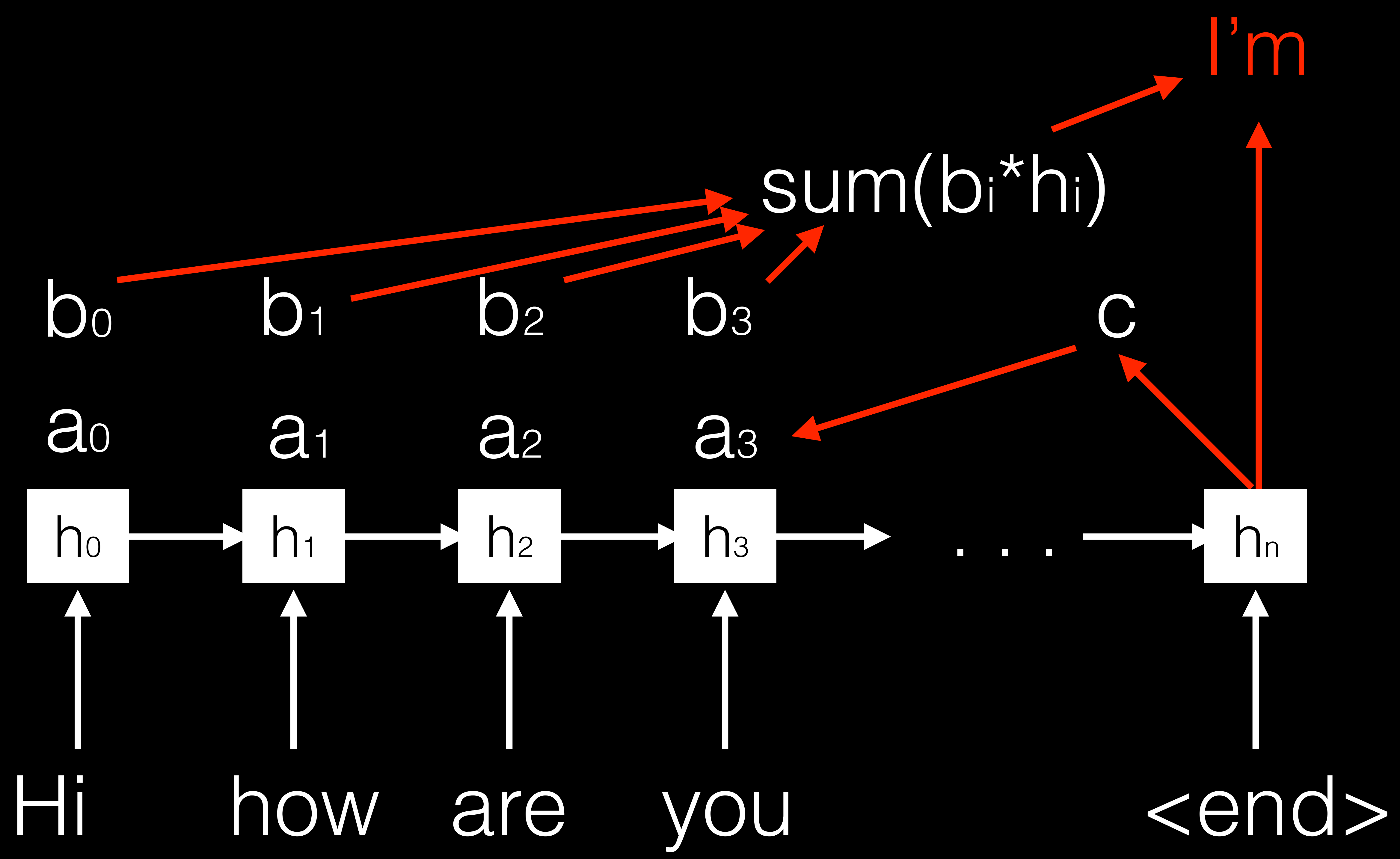


Attention Mechanism



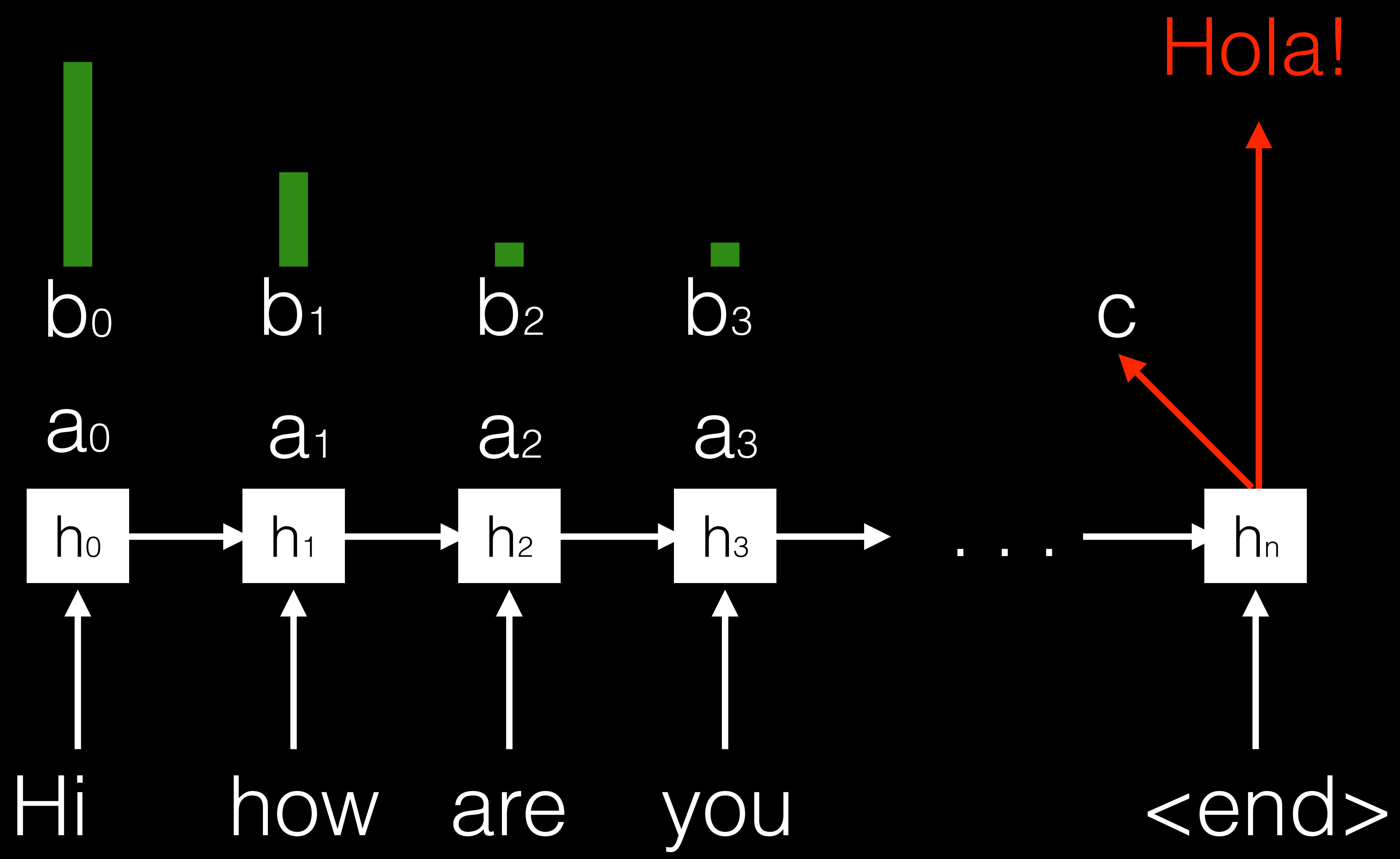
$$b_i = \frac{\exp(a_i)}{\exp(a_1) + \exp(a_2) + \ldots + \exp(a_n)}$$

Attention Mechanism

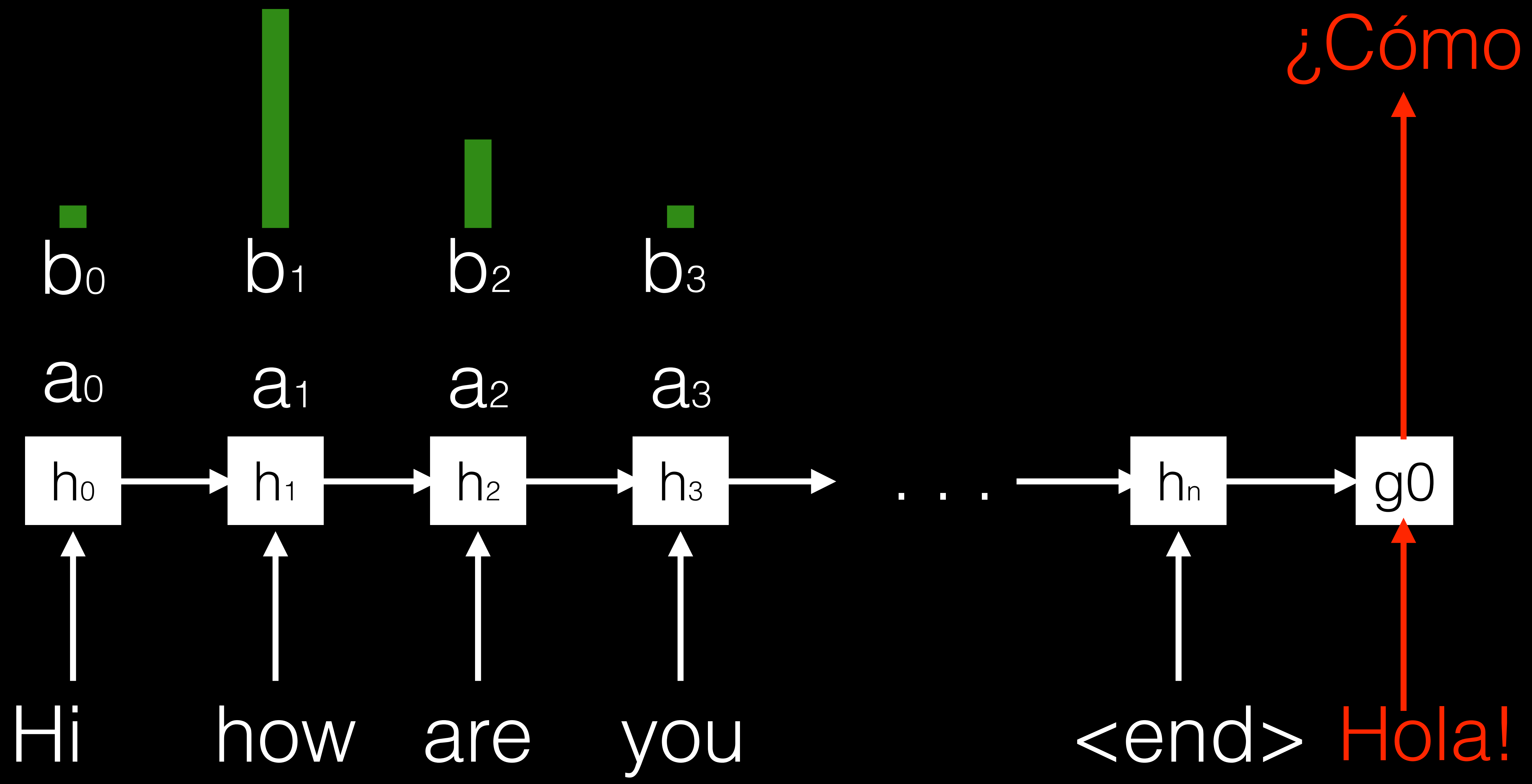


Implemented in TensorFlow seq2seq

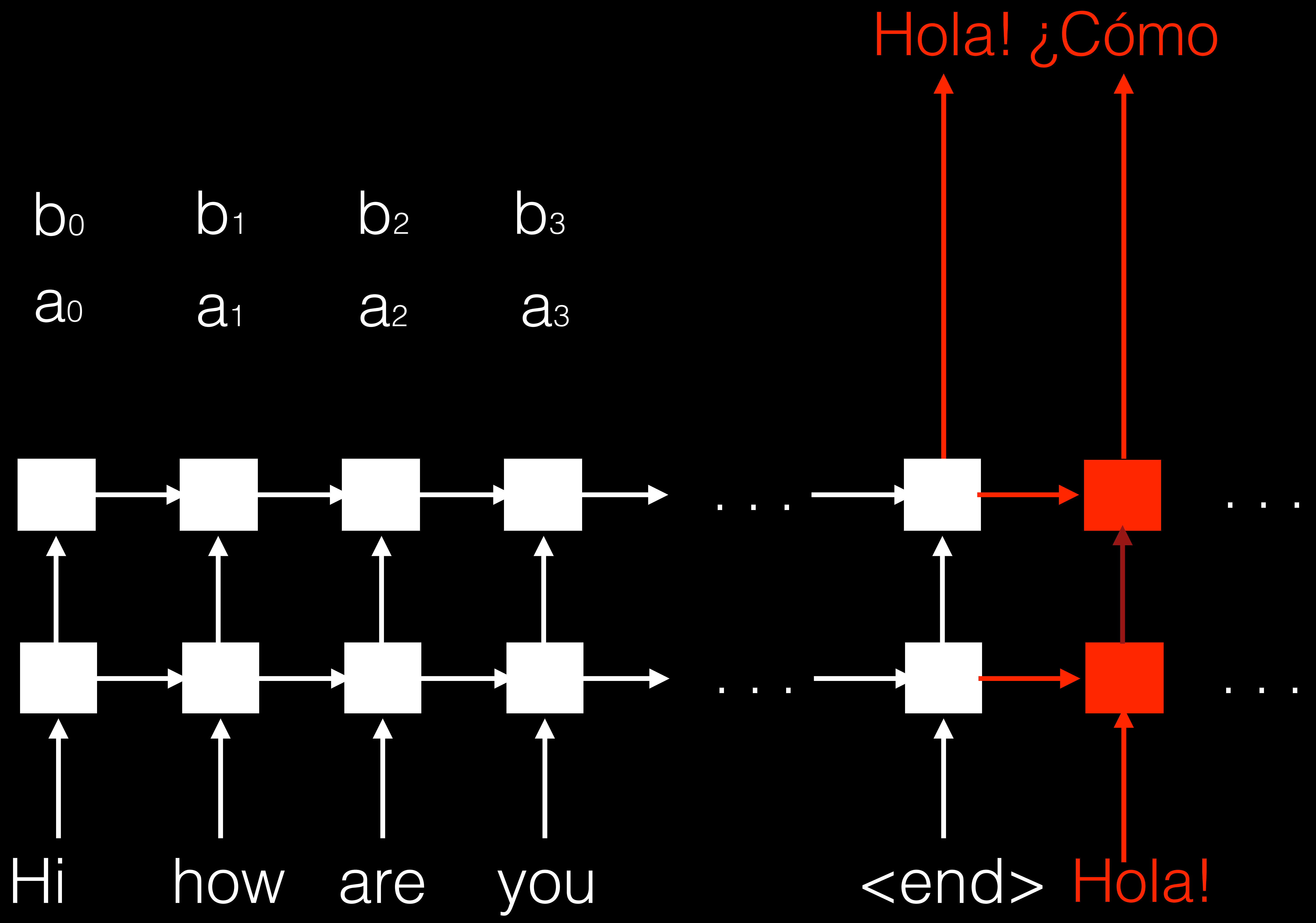
Model Understandability with Attention Mechanism



Model Understandability with Attention Mechanism



Deeper Networks work Better



Sequence to Sequence With Attention

- Currently the state-of-art in many translation tasks
    - Tip 1: Use word segments or word/character hybrid instead of just words
    - Tip 2: Gradient Clipping to prevent explosion
    - Tip 3: Use Long Short Term Memory

LSTMCell vs. RNNCell

RNNCell:

```
h = tanh(theta * [inputs, h])
```

LSTMCell:

```
Z = theta * [inputs, h]

i, j, f, o = split(1, 4, Z) # split to four blocks

new_c = c * sigmoid(f) + sigmoid(i) * tanh(j) # integral of c

new_h = tanh(new_c) * sigmoid(o)
```

Applications

- Other applications:
  - Summarization, Image Captioning,
  - Speech Transcription, Q&A

Applications

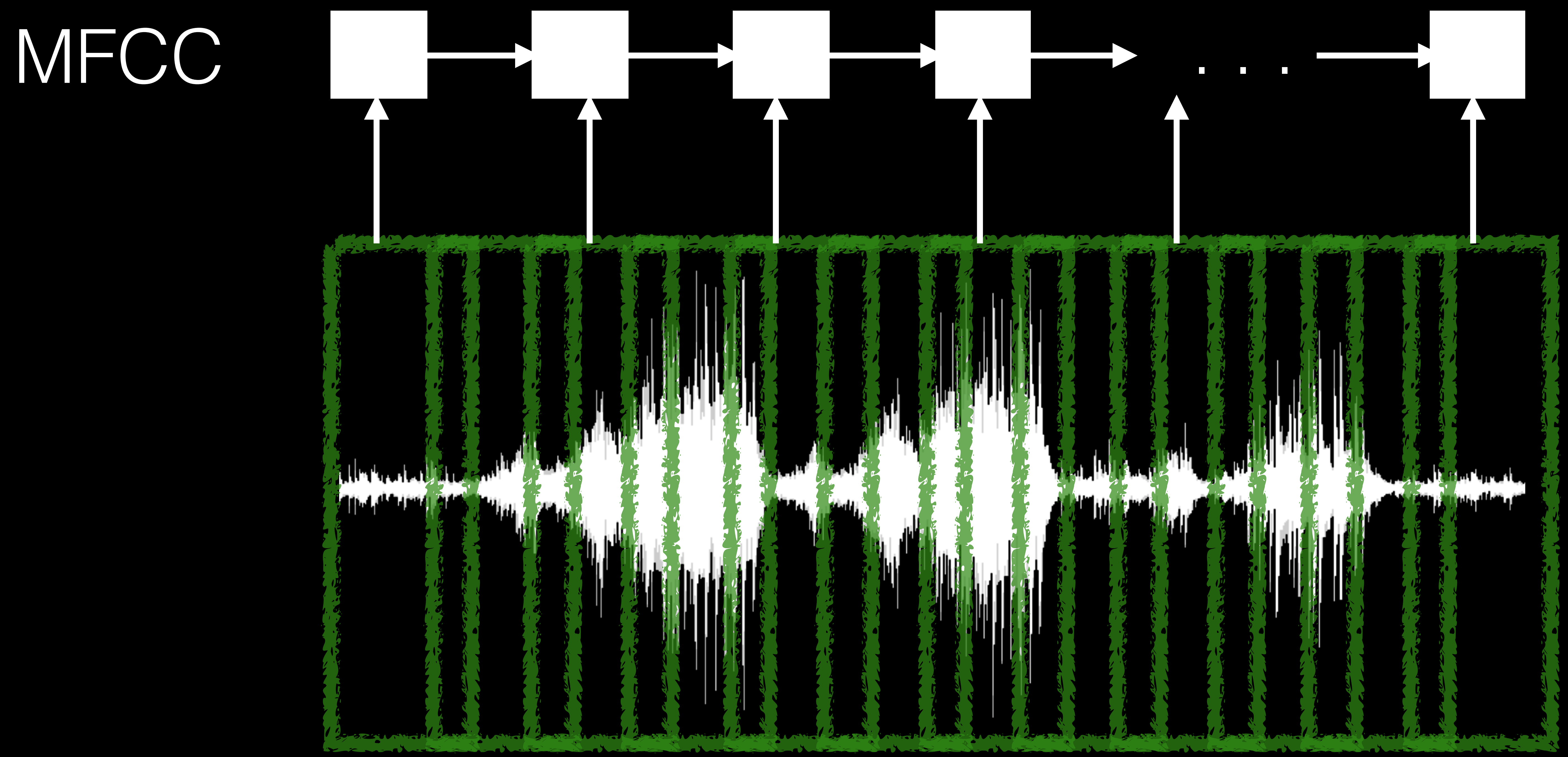
- Other applications:
  - Summarization, Image Captioning,
  - Speech Transcription, Q&A

seq2seq for Speech

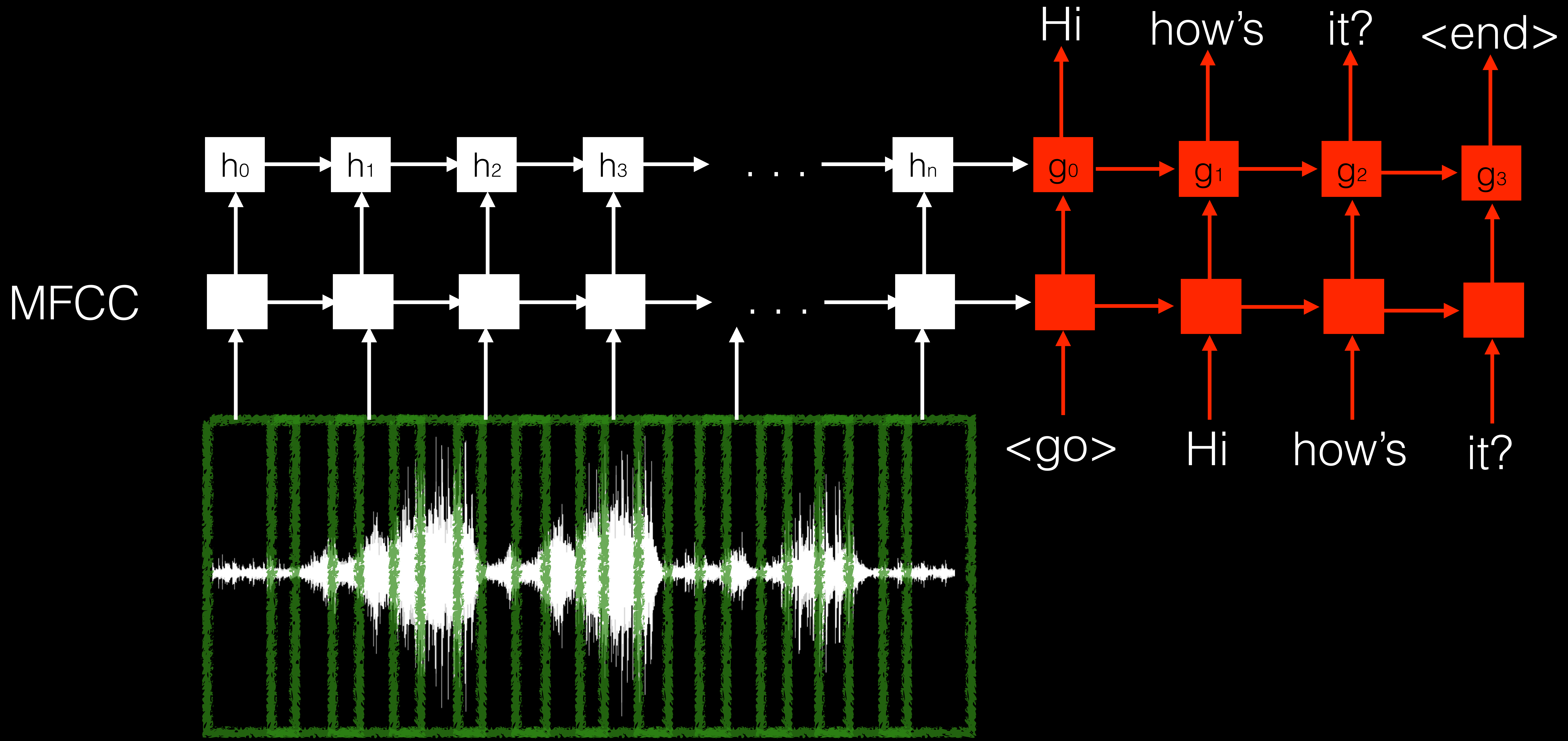
—> Hi how’s it?

seq2seq for Speech

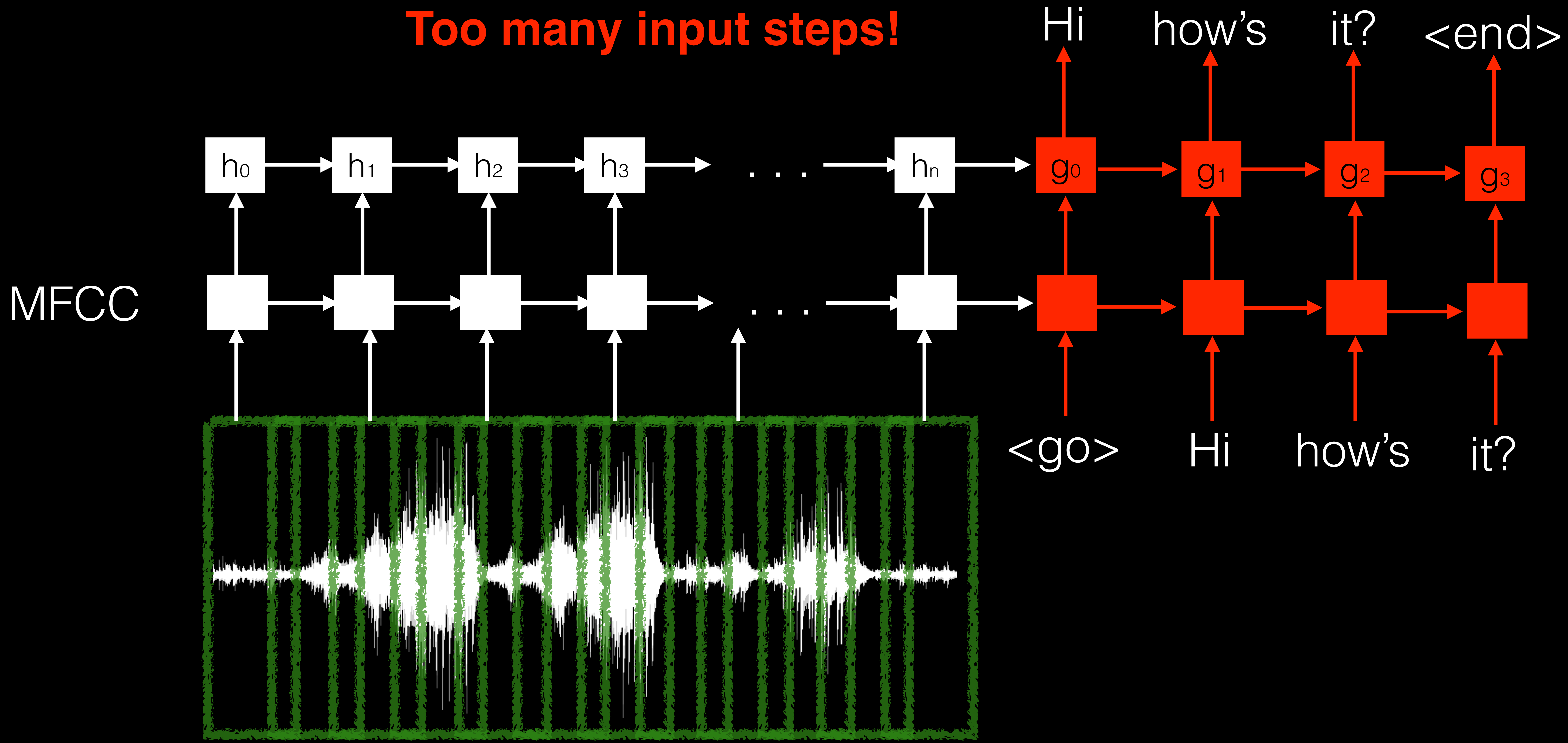
seq2seq for Speech



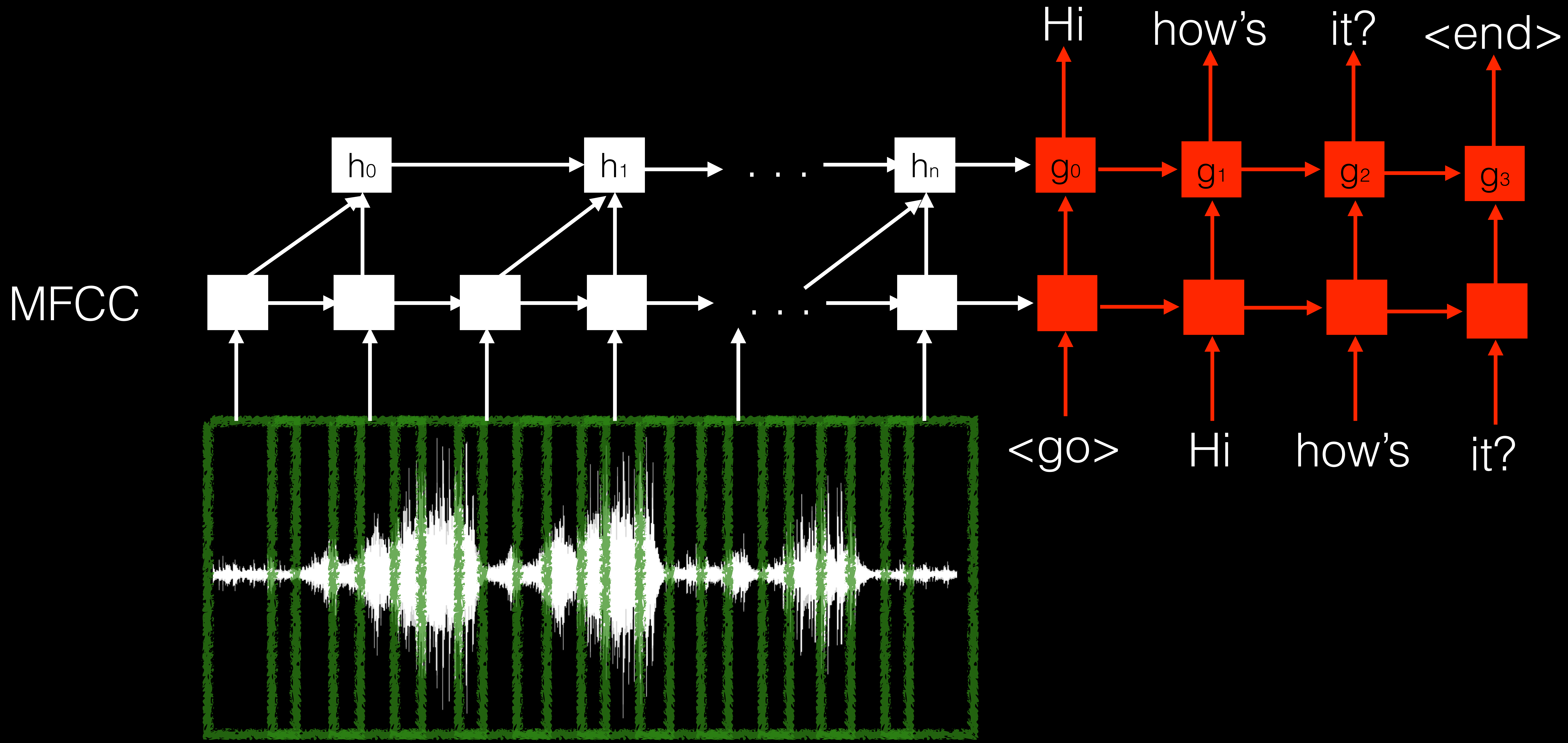
seq2seq for Speech



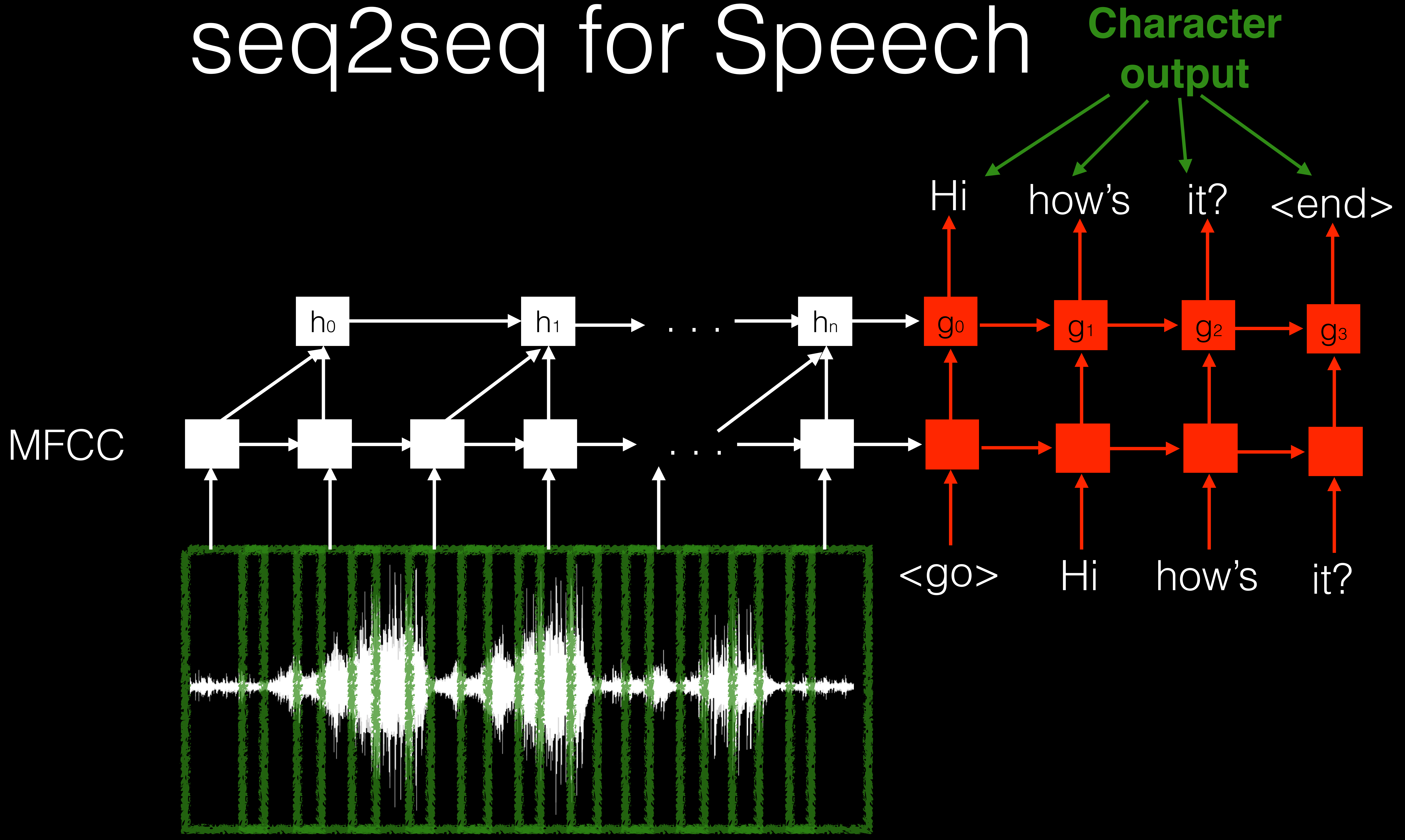
seq2seq for Speech



seq2seq for Speech



seq2seq for Speech



Sequence to Sequence With Attention for Speech

- Implicit language model
- “Offline” beam search decoding
- Not as good as
  - CTC (Adam Coates’ talk)
  - HMM-DNN hybrid (most widely-used speech systems)

The Big Picture

- Sequence to sequence is an “End-to-end Deep Learning” algorithm
- It’s very general, so it should work with most NLP-related tasks **when you have a lot of data**
- If you don’t have enough data:
  - Consider dividing your problem into smaller problems, and train seq2seq on each of them.
  - **Train jointly with many other tasks**
- What I present next is an active area of research

Automatic Q&A

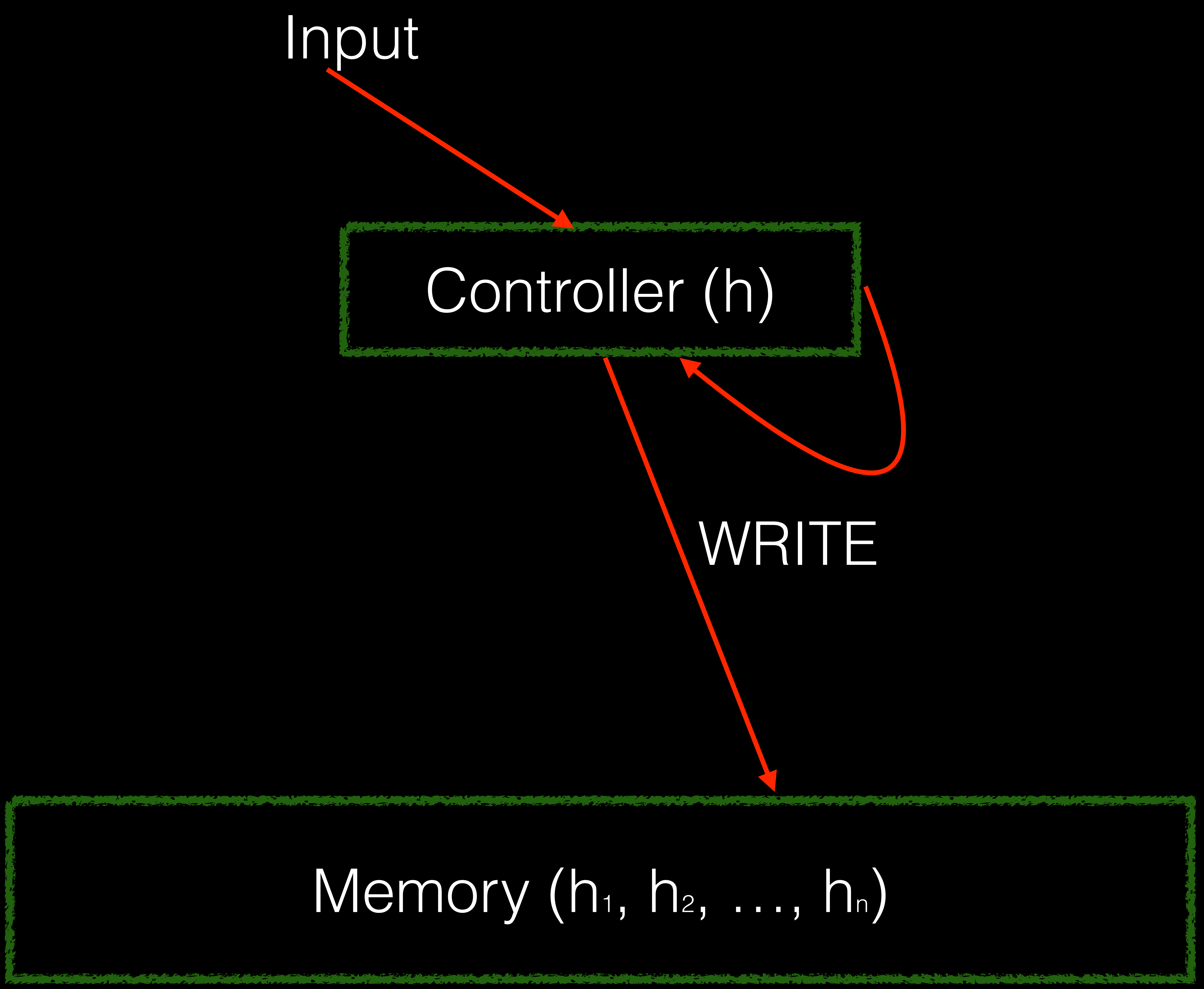
- Reading a book and answer a question

- Seq2seq with attention: Read the book, then read the question, then revisit all pages in the book.

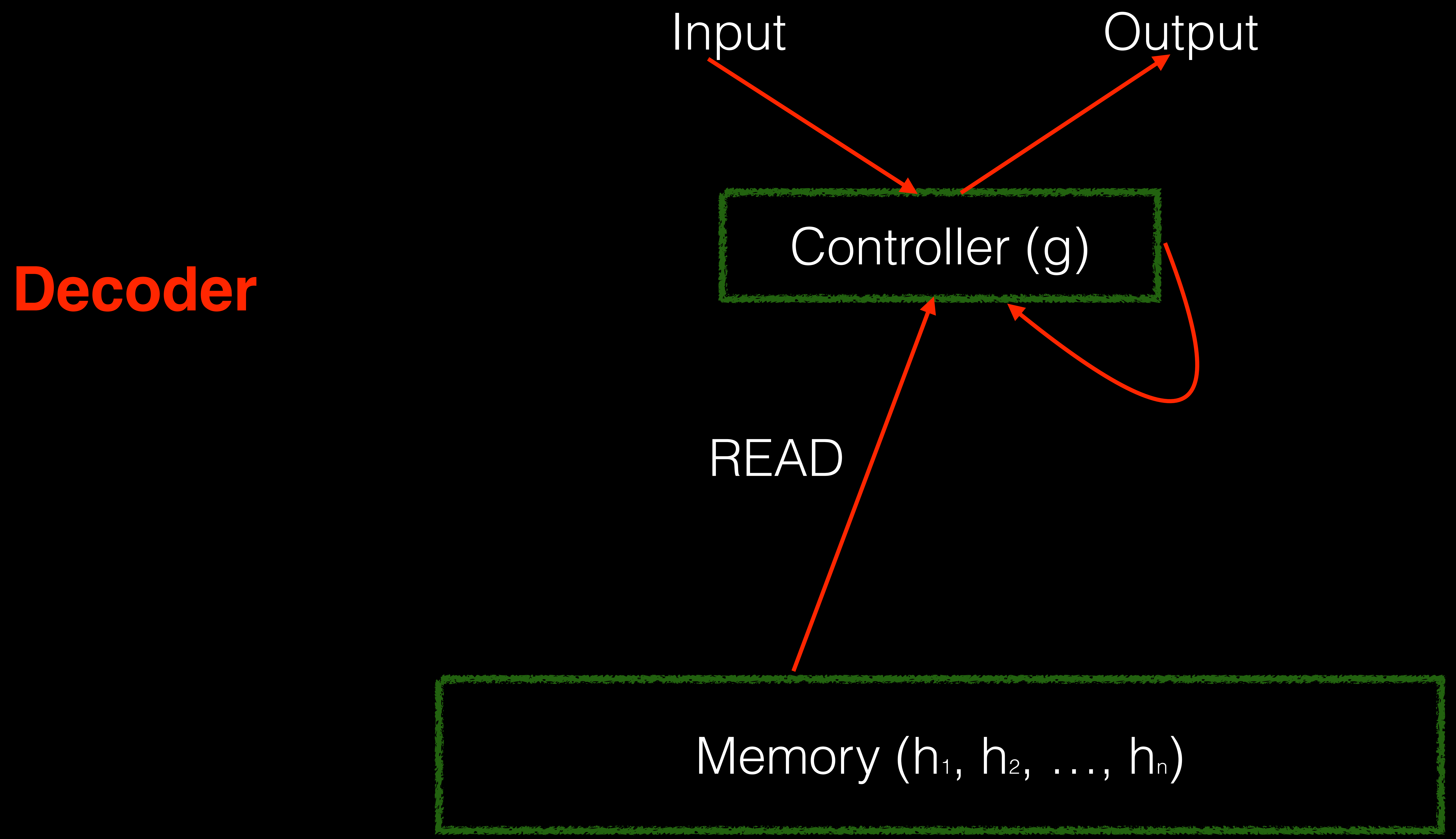
—> Augmented RNNs with memory (Memory Networks, Neural Turing Machines, Dynamic Memory Networks, Stack-augmented RNNs etc.)

Revisit Attention Mechanism

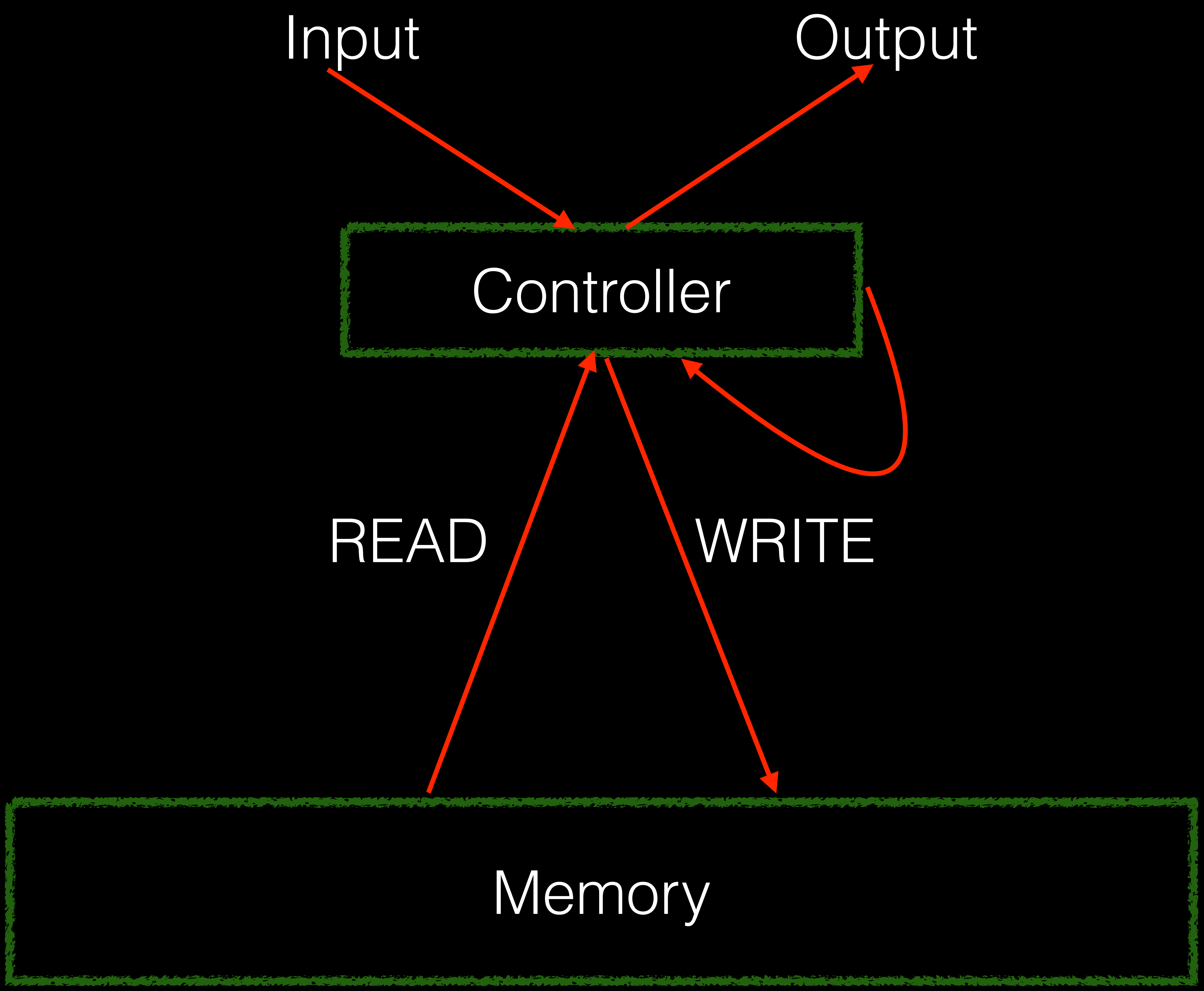
Encoder



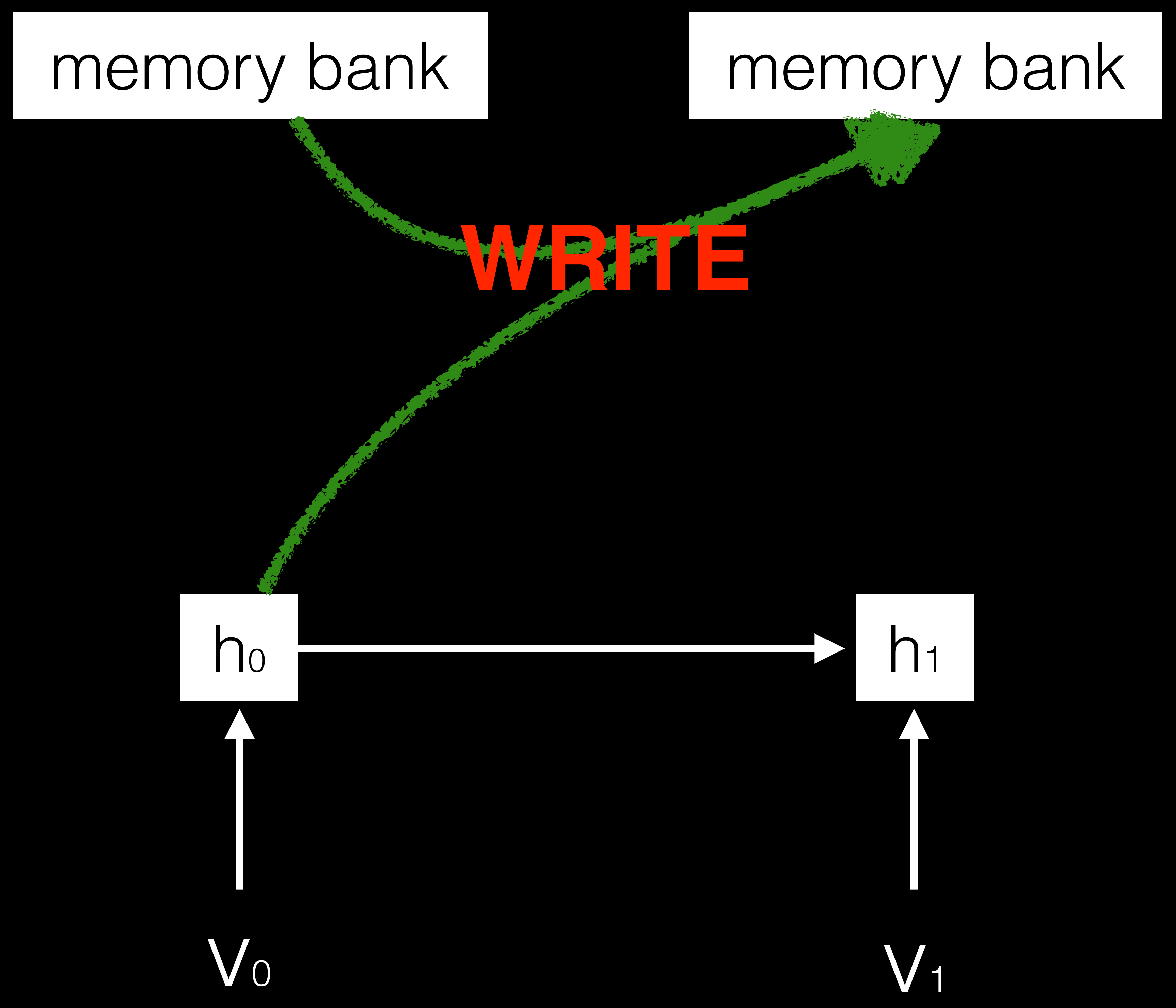
Revisit Attention Mechanism



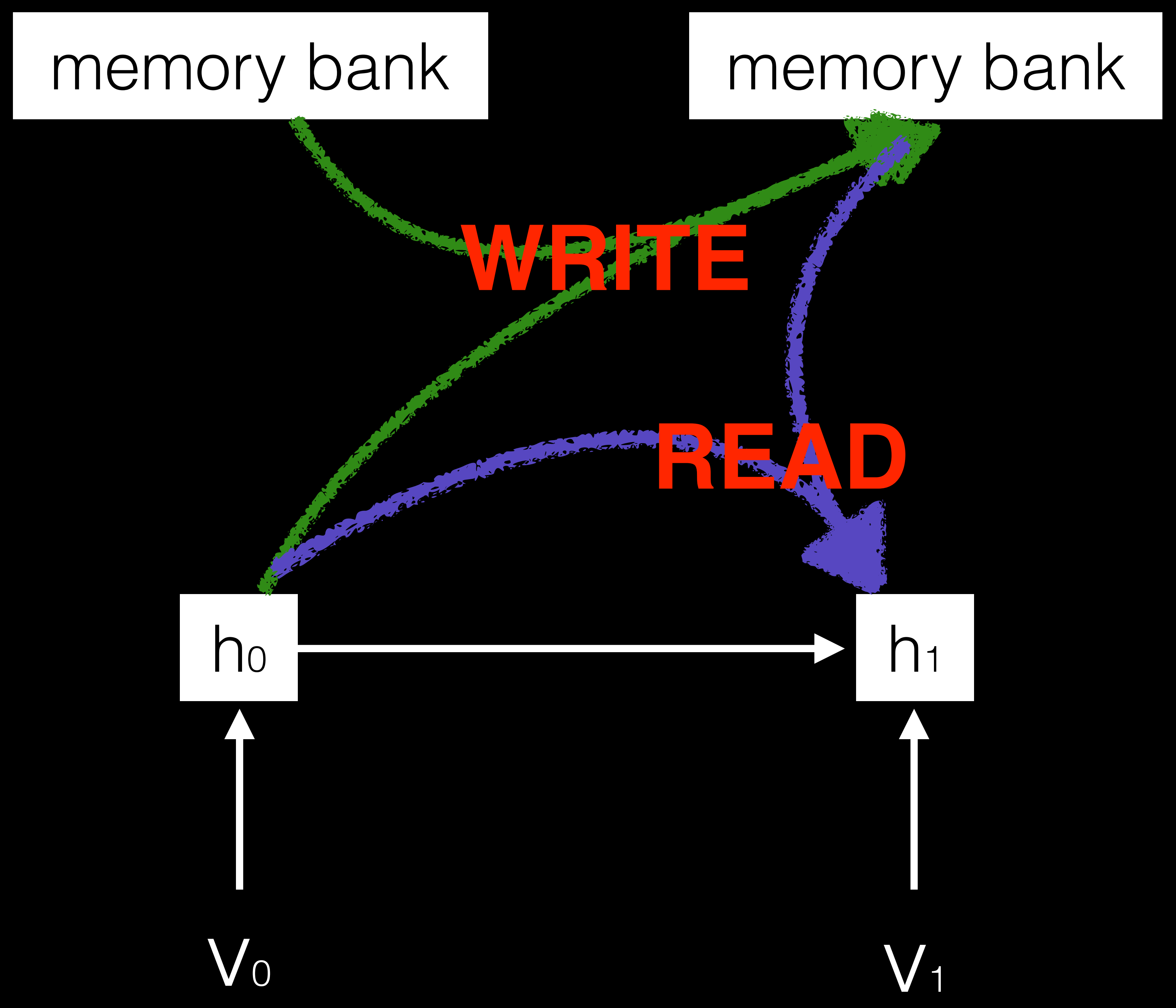
Differentiable Memory (Neural Turing Machines, Memory Networks, Stack-Augmented RNNs)



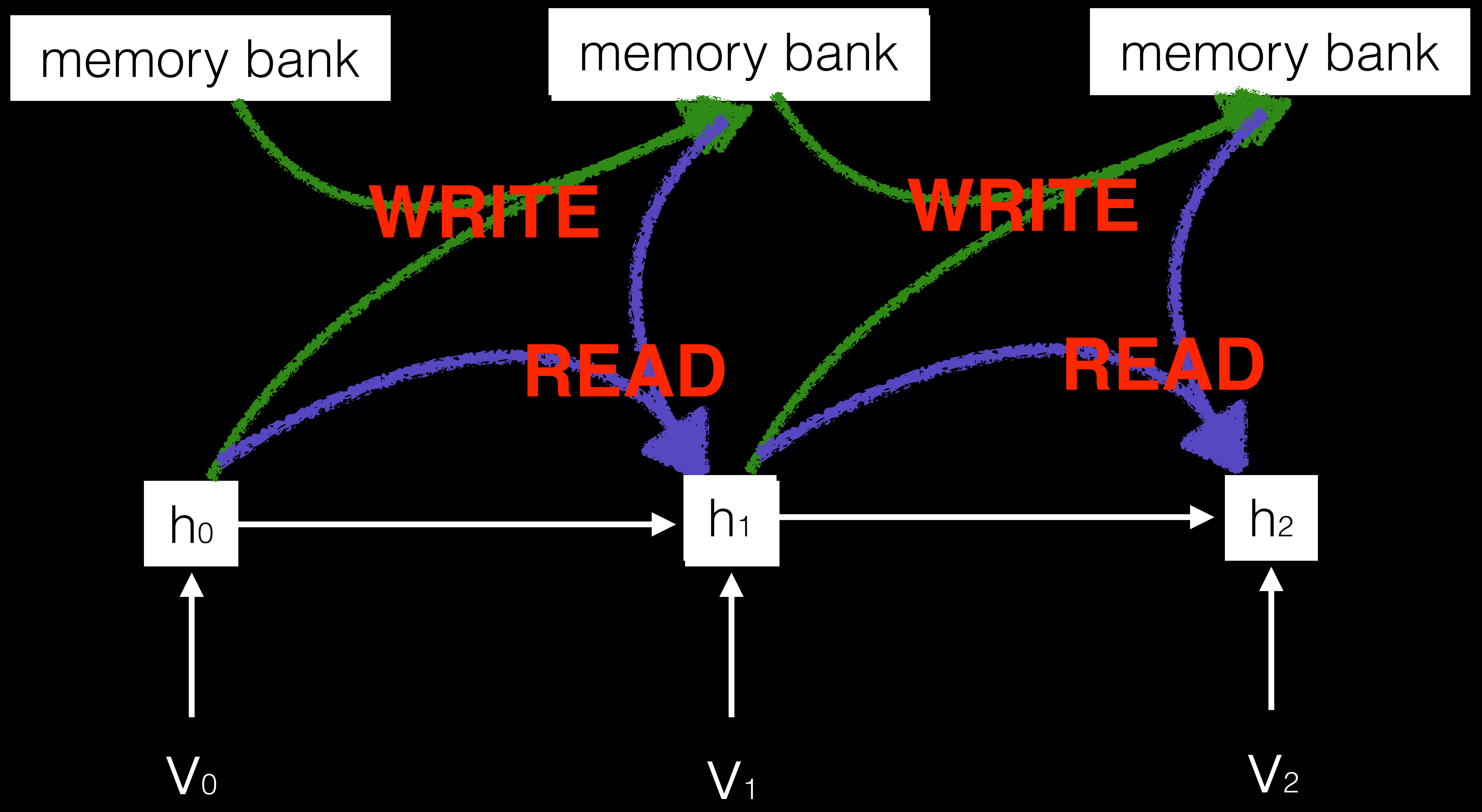
Differentiable Memory



Differentiable Memory



Differentiable Memory

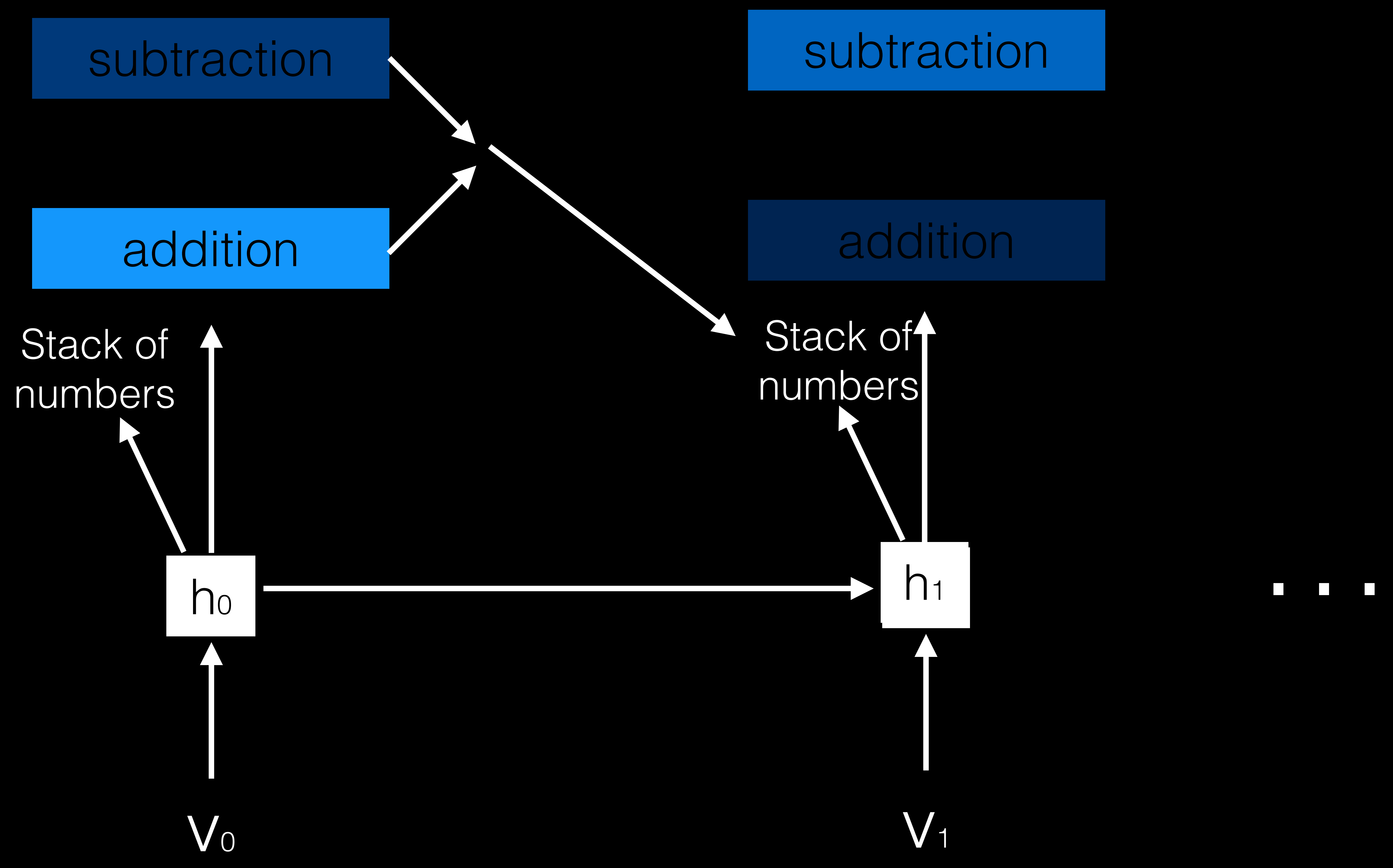


RNN with augmented memory

RNN with augmented operations

- Context: The building was constructed in 2000 . … It was destroyed in 2010 ….
- Question: How long did the building survive?
- Answer: 10 years .

Neural Programmers



The Big Picture

- Sequence to sequence is an “End-to-end Deep Learning” algorithm
- It’s very general, so it should work with most NLP-related tasks **when you have a lot of data**
- If you don’t have enough data:
  - Consider dividing your problem into smaller problems, and train seq2seq on each of them.
  - **Train jointly with many other tasks**
- RNN with memory, or operation augmentation are exciting work in progress

Additional Reading

- Chris Olah's blog: Attention and Augmented Recurrent Neural Networks
- My own tutorials: http://ai.stanford.edu/~quocle/tutorial2.pdf
- Seq2seq in TensorFlow: https://www.tensorflow.org/versions/r0.10/tutorials/seq2seq/index.html

References

- Modeling
  - Sequence to Sequence with Neural Networks by Sutskever, Vinyals, Le. NIPS, 2014
  - Neural machine translation by jointly learning to align and translate by Bahdanau, Cho, Bengio. ICLR, 2015
  - Neural Turing Machines, by Graves, Wayne, Danihelka. arXiv, 2014
  - End-to-End Memory Networks by Sukhbaatar, Weston, Fergus. NIPS, 2015
  - Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks, by Bengio, Vinyals, Jaitly, Shazeer. NIPS, 2015
  - Inferring Algorithmic Patterns with Stack-Augmented Recurrent Nets by Joulin and Mikolov. NIPS, 2015.
- Applications
  - Show and Tell: A Neural Image Caption Generator, by Vinyals, Toshev, Bengio, Erhan. CVPR, 2015
  - Grammar as Foreign Language by Vinyals, Kaiser, Koo, Petrov, Sutskever, Hinton. NIPS, 2015
  - Neural Conversational Model, by Vinyl and Le. ICML Workshop, 2015
  - A neural network approach to context-sensitive generation of conversational responses, by Sordoni, Galley, Auli, Brockett, Ji, Mitchell, Gao, Dolan, Nie. NAACL, 2015.
  - Neural responding machine for short-text conversation by Shang, Lu, Li. ACL, 2015.
  - Attention-Based Models for Speech Recognition. Chorowski, Bahdanau, Serdyuk, Cho, Bengio. NIPS, 2015
  - Listen, Attend and Spell. Chan, Jaitly, Le, Vinyals. ICASSP, 2016