CS 581

Advanced Artificial Intelligence

April 3, 2024

Announcements / Reminders

- **Please follow the Week 11/12 To Do List instructions (if you haven't already)**

- **Programming Assignment #02 due on Sunday (04/07) at 11:59 PM CST**

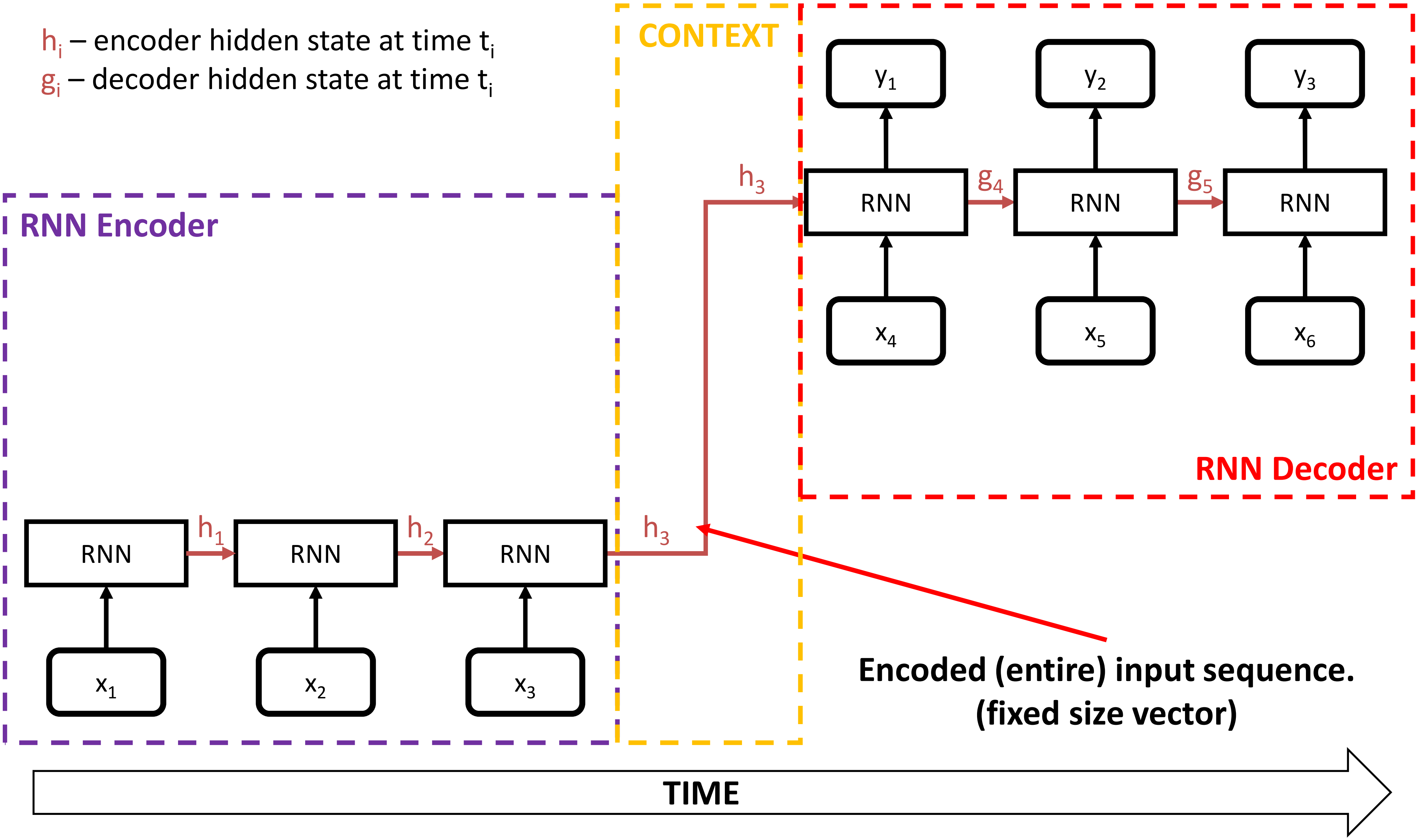
Plan for Today

- **Attention Mechanism**
- **Transformer Basics**
- **Generative AI Models: Introduction**

Sequence to Sequence Networks (seq2seq) With Attention

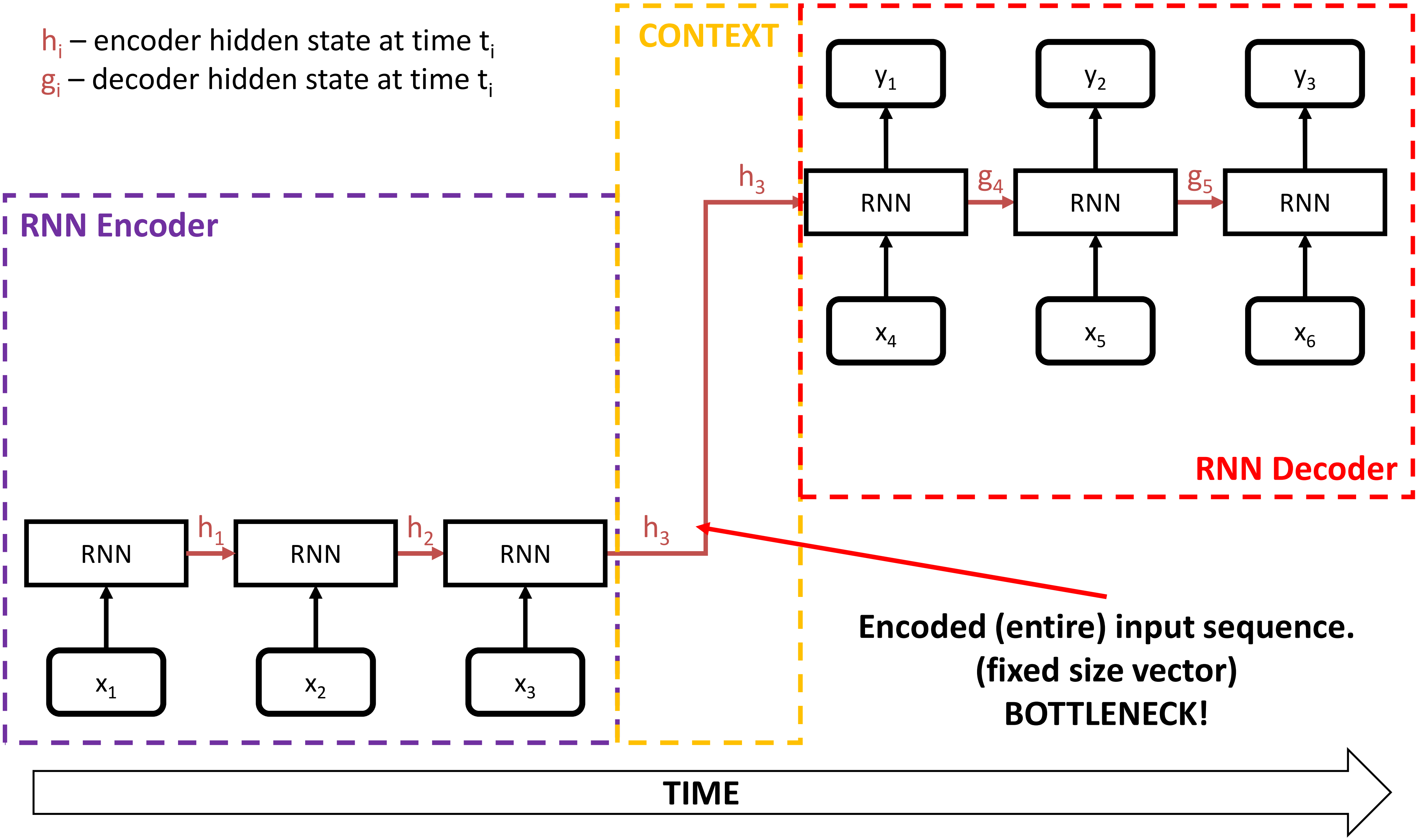
RNN Encoder-Decoder Architecture

h_i – encoder hidden state at time t_i
g_i – decoder hidden state at time t_i



RNN Encoder-Decoder: Context

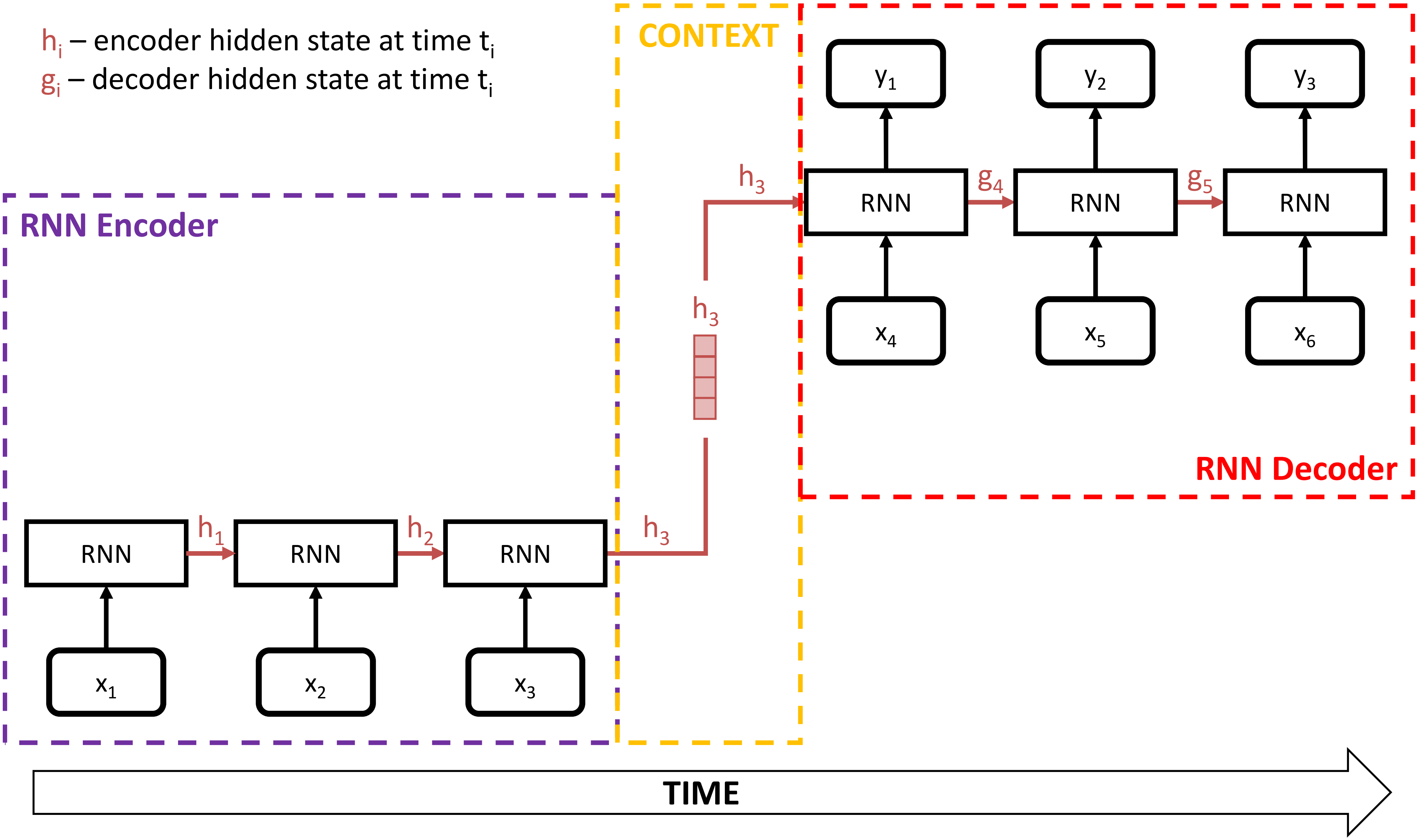
h_i – encoder hidden state at time t_i
g_i – decoder hidden state at time t_i



Encoded (entire) input sequence.
(fixed size vector)
BOTTLENECK!

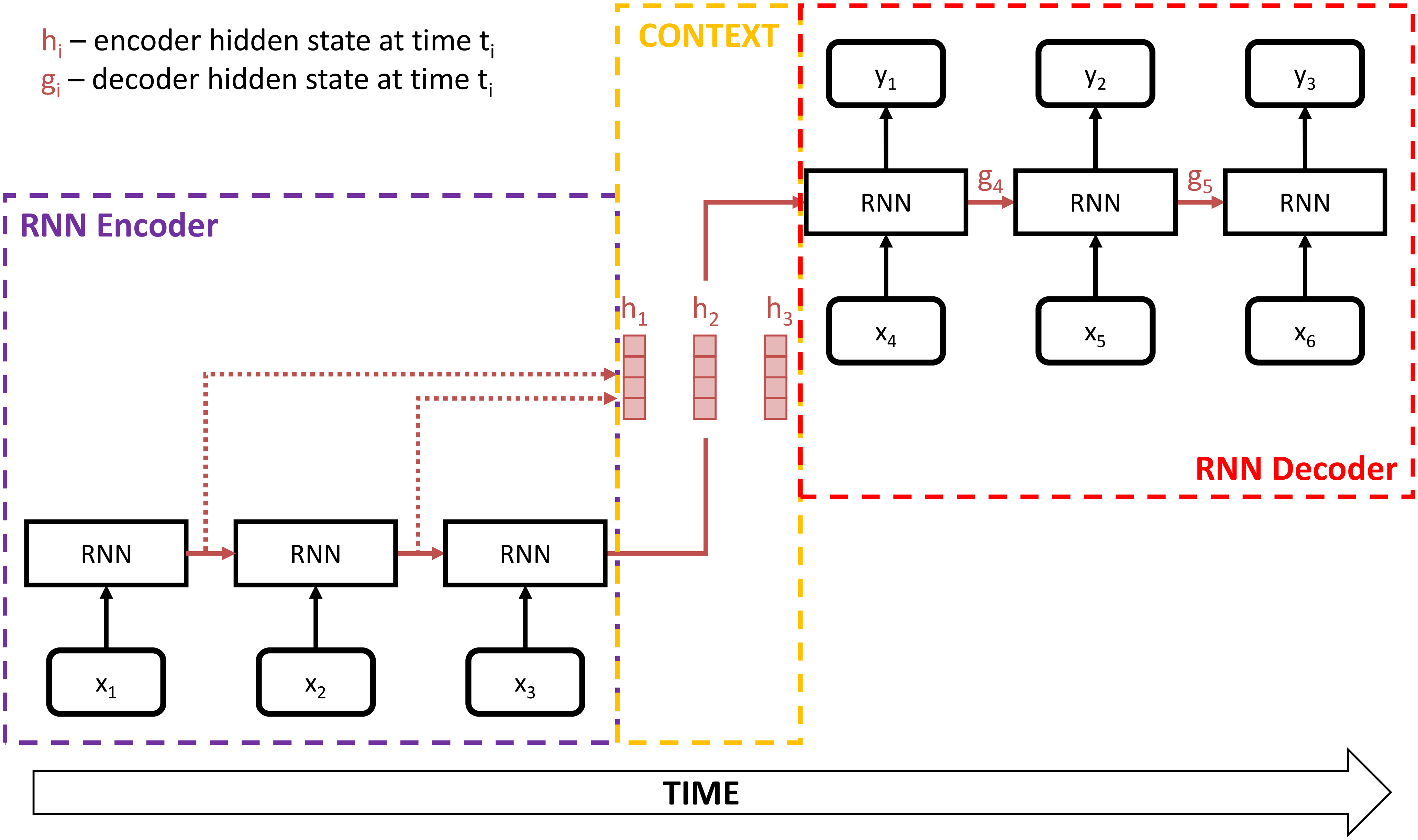
Fixed Length Context

h_i – encoder hidden state at time t_i
g_i – decoder hidden state at time t_i



RNN Encoder-Decoder Architecture

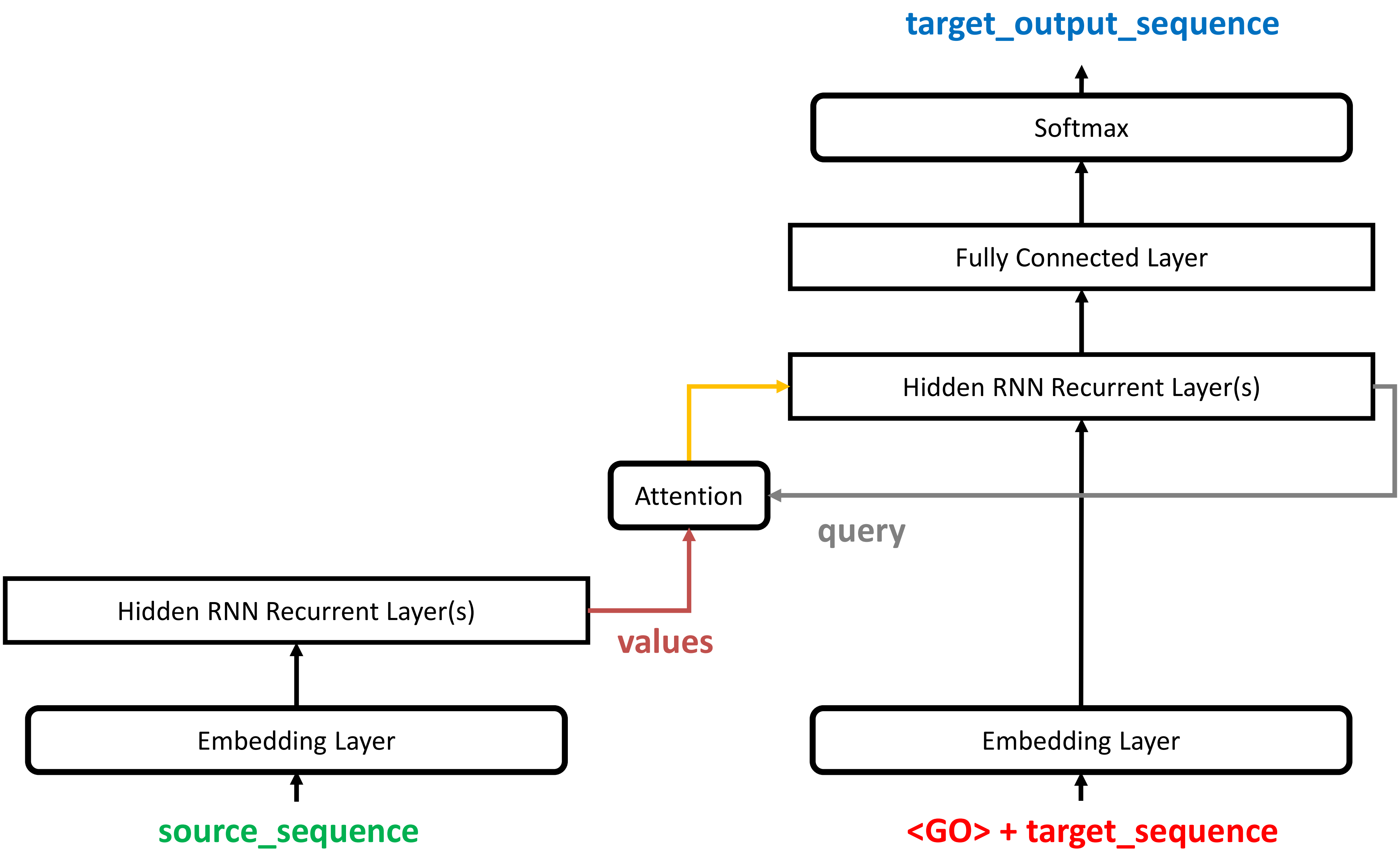
h_i – encoder hidden state at time t_i
g_i – decoder hidden state at time t_i



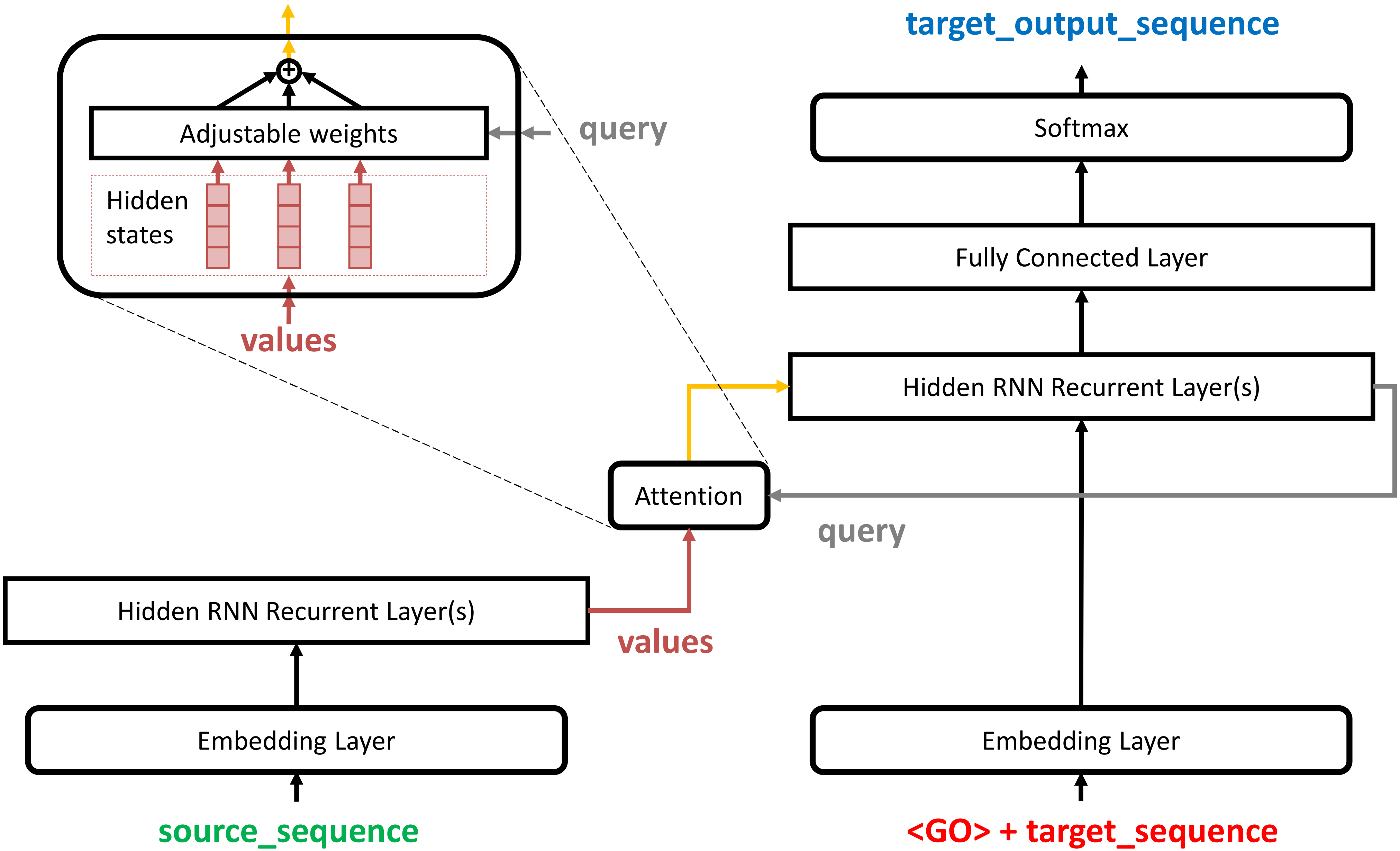
Attention Mechanism

- Given a set of vector values, and a vector query, attention is a technique to compute a weighted sum of the values, dependent on the query
- Attention mechanism “amplifies” important aspects of the signal from the encoder based on the decoder query
- In seq2seq models with attention, each decoder hidden state (query) attends to all the encoder hidden states (values)

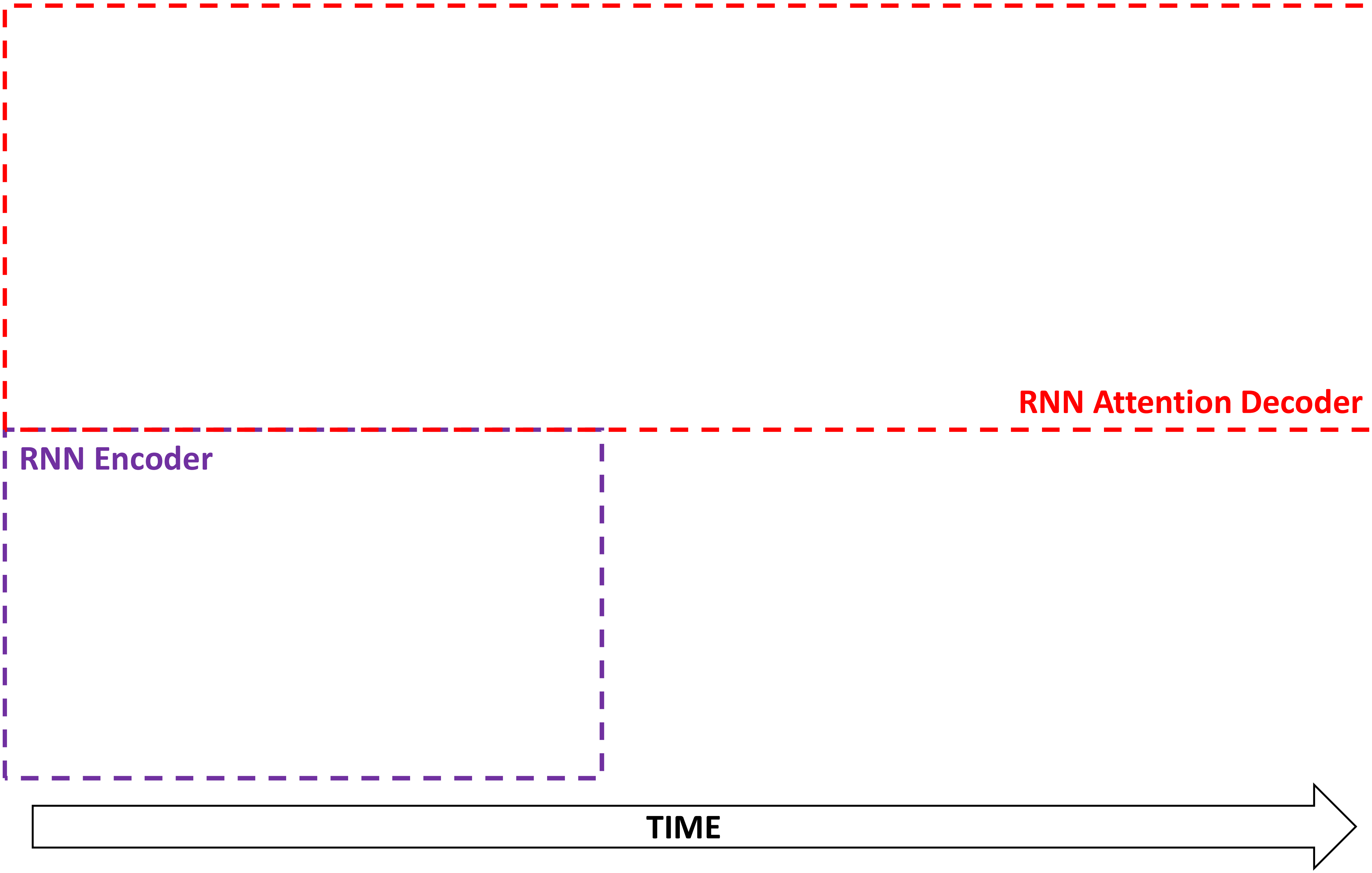
RNN Encoder-Decoder with Attention



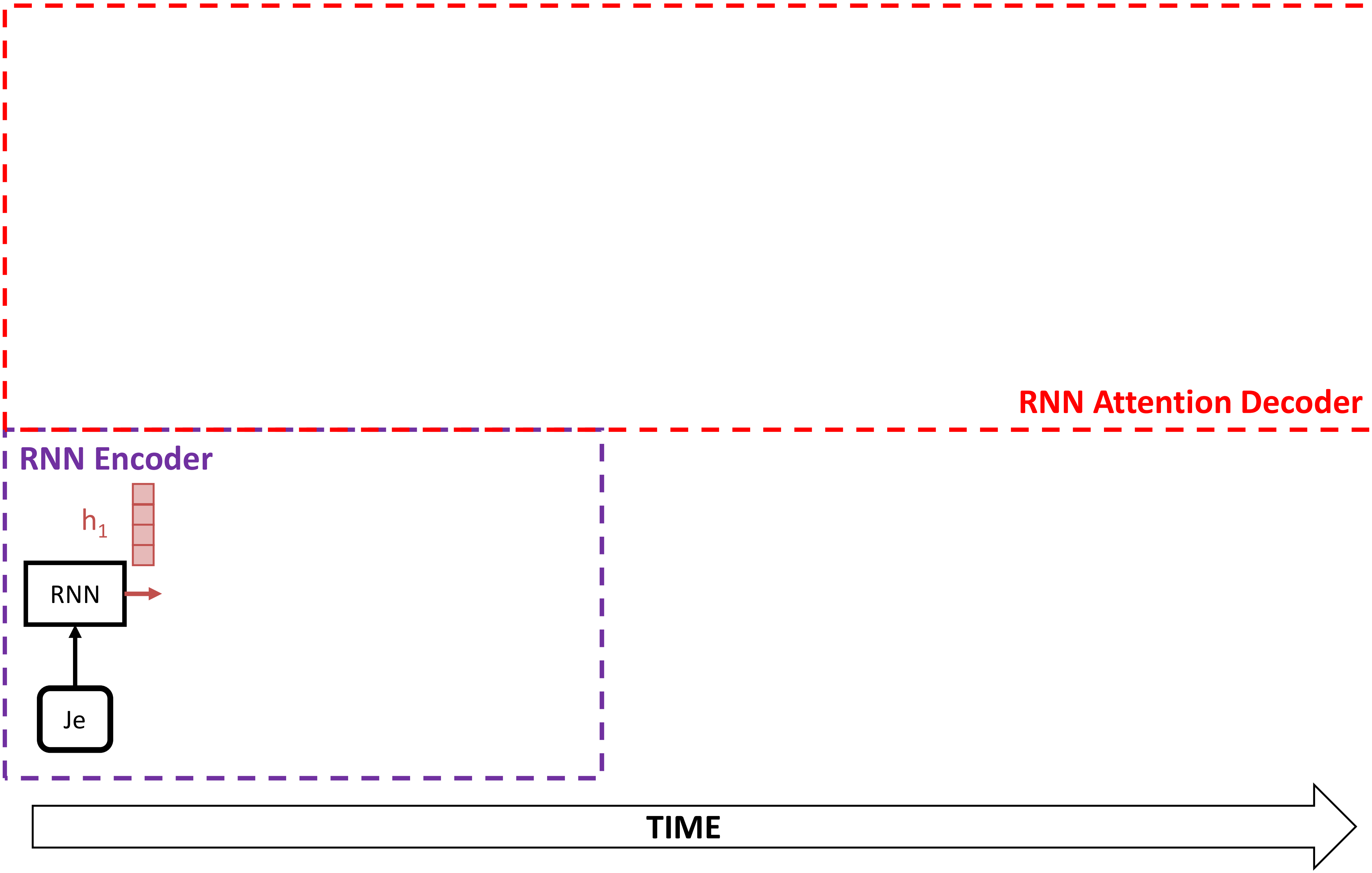
RNN Encoder-Decoder with Attention



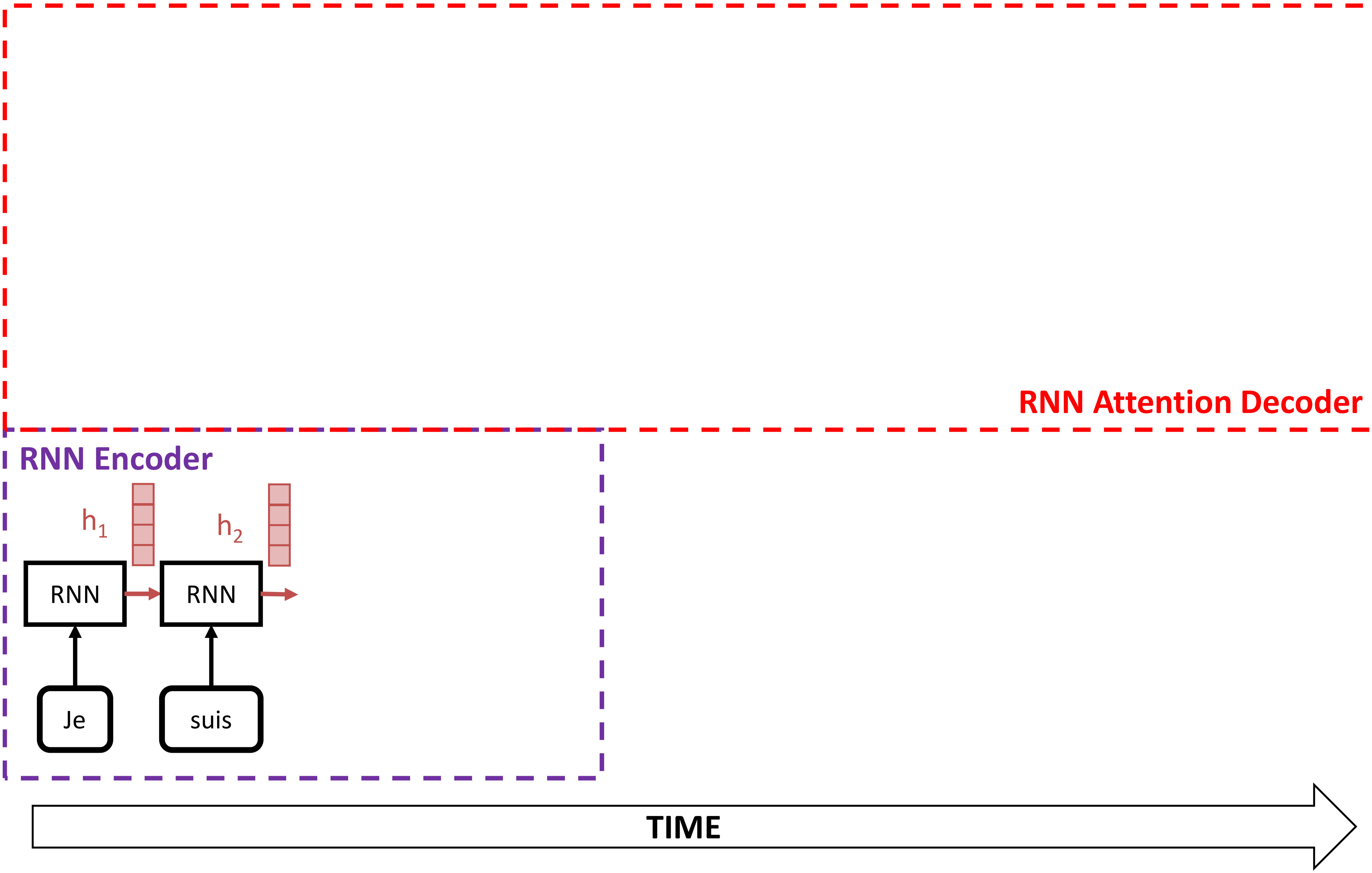
RNN Encoder-Decoder with Attention



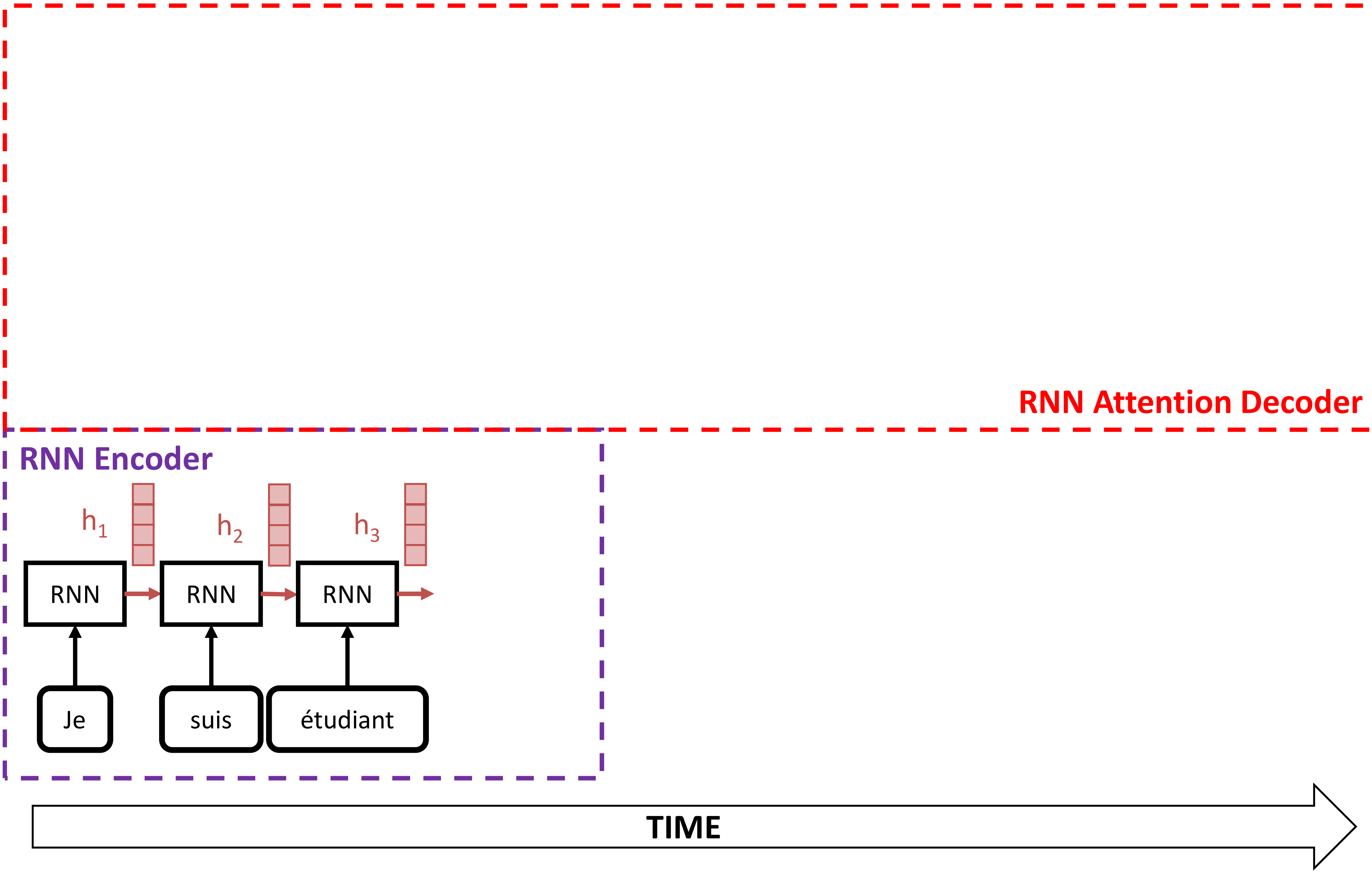
RNN Encoder-Decoder with Attention



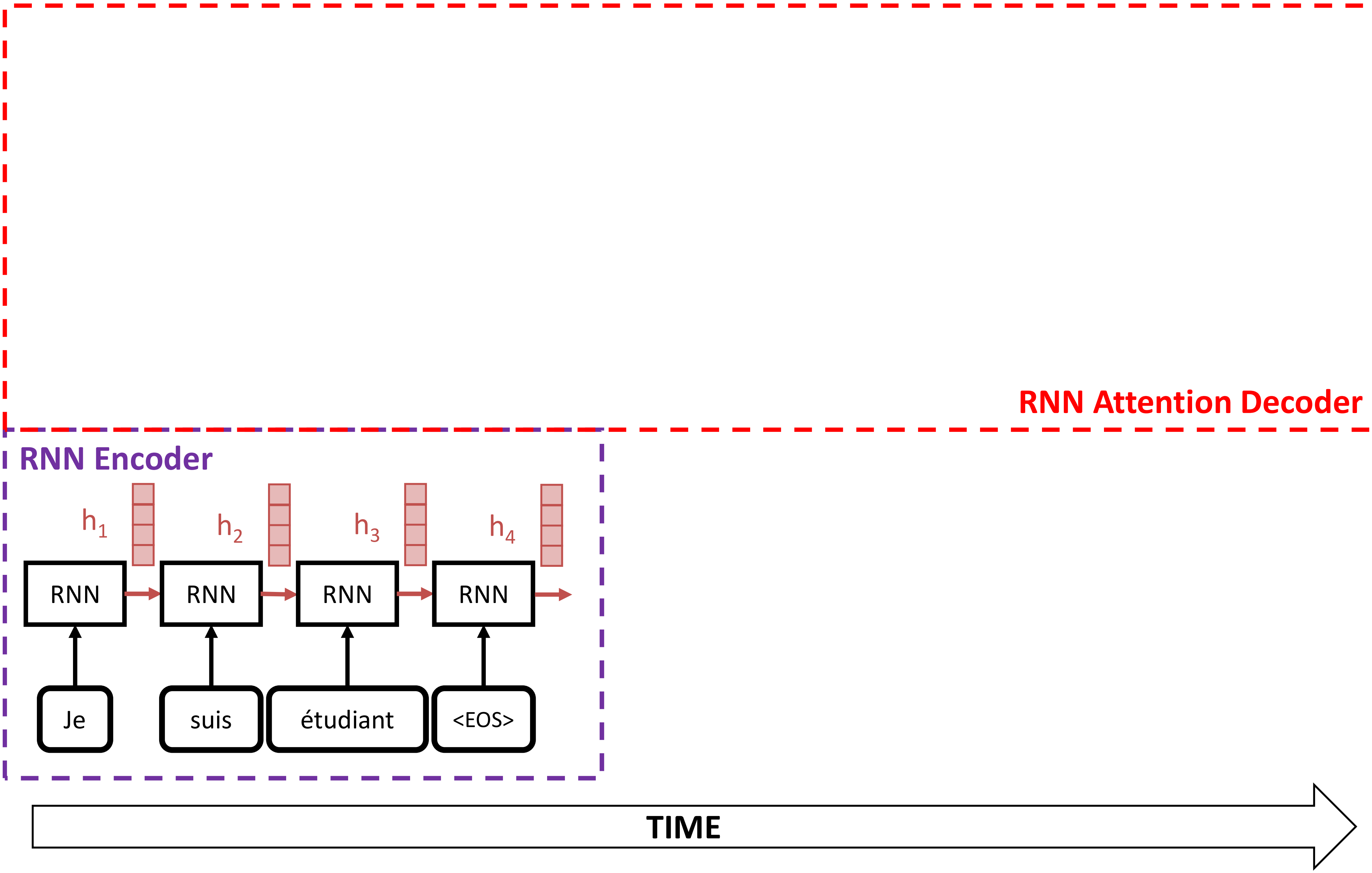
RNN Encoder-Decoder with Attention



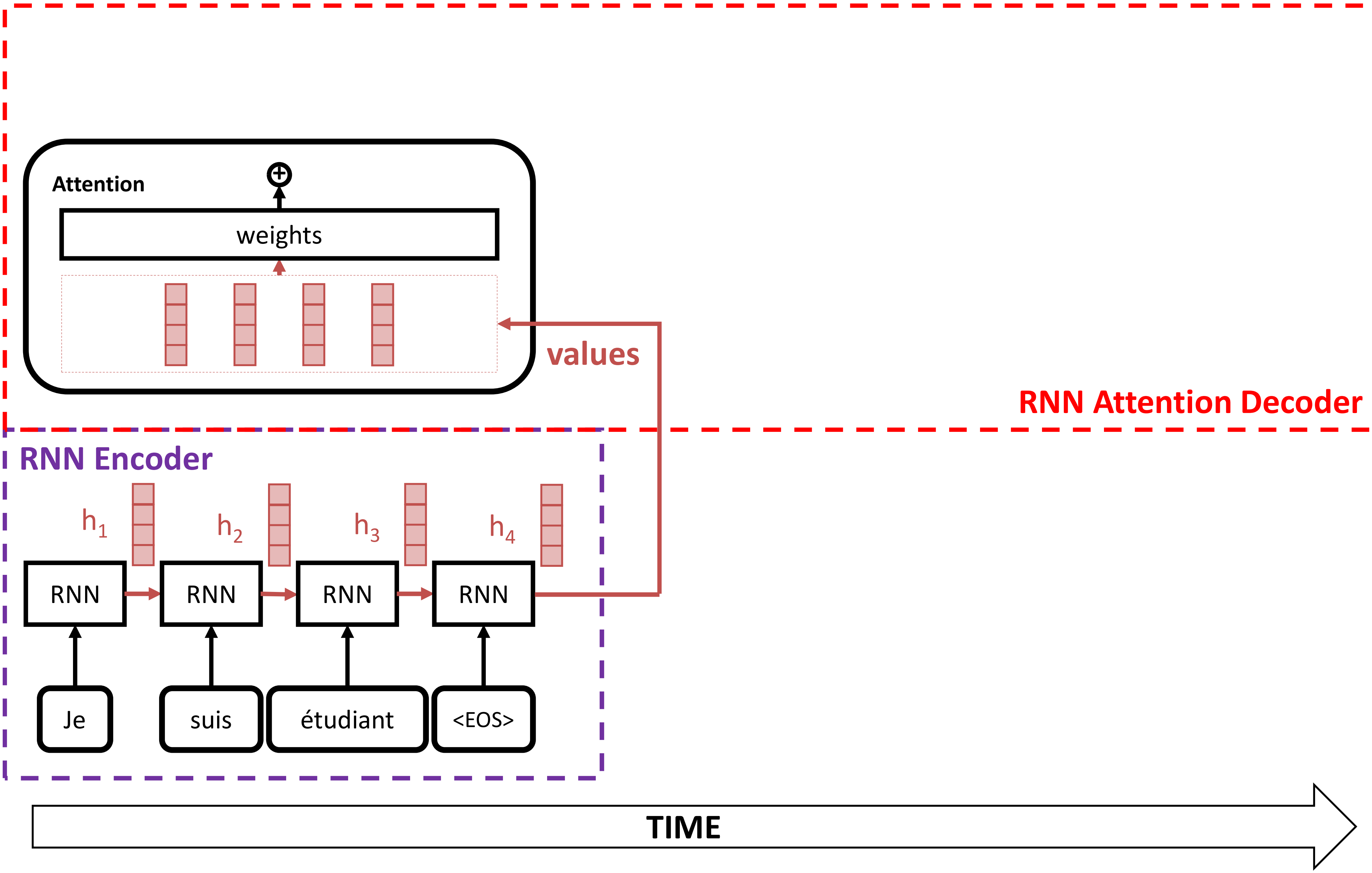
RNN Encoder-Decoder with Attention



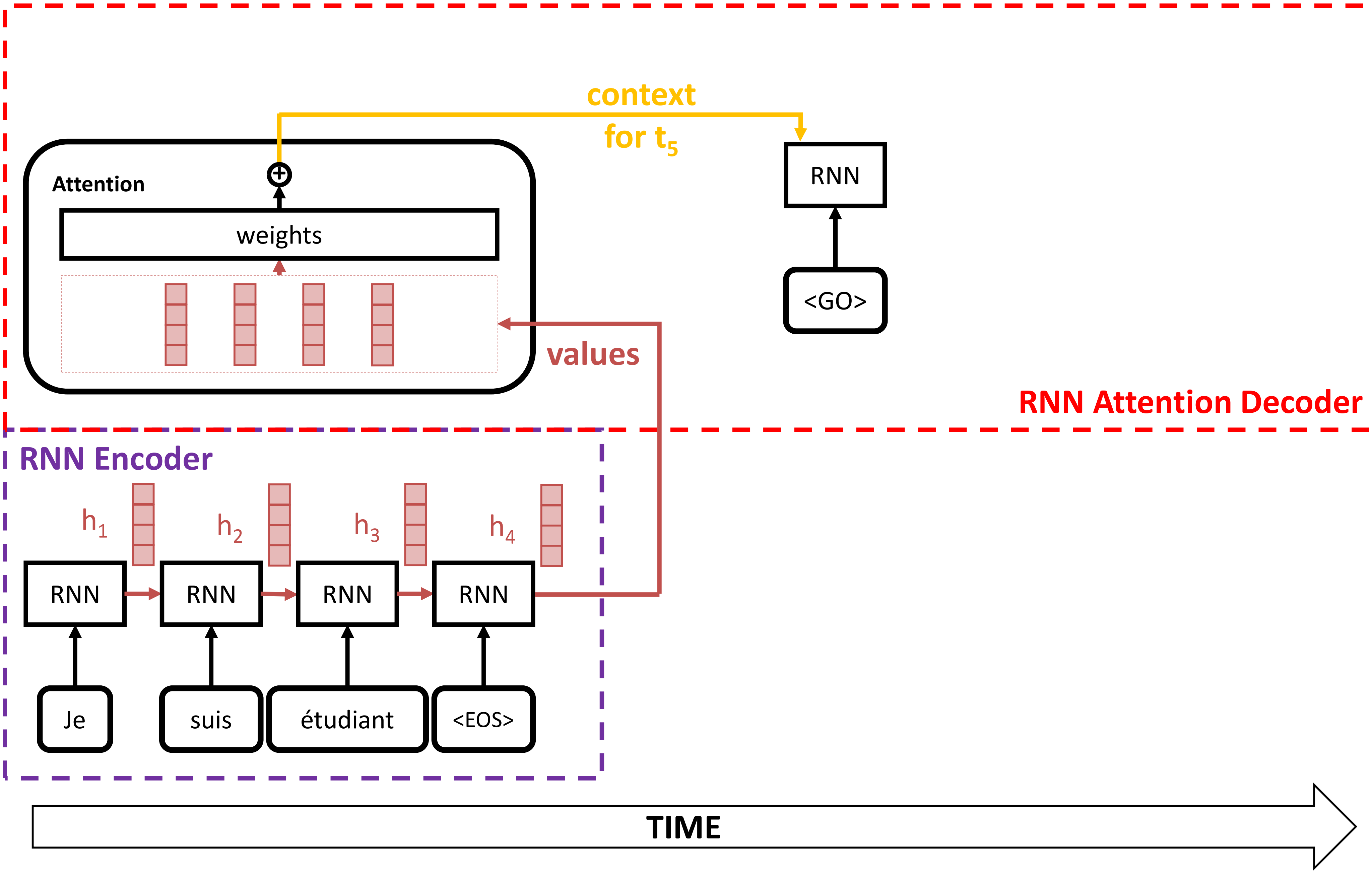
RNN Encoder-Decoder with Attention



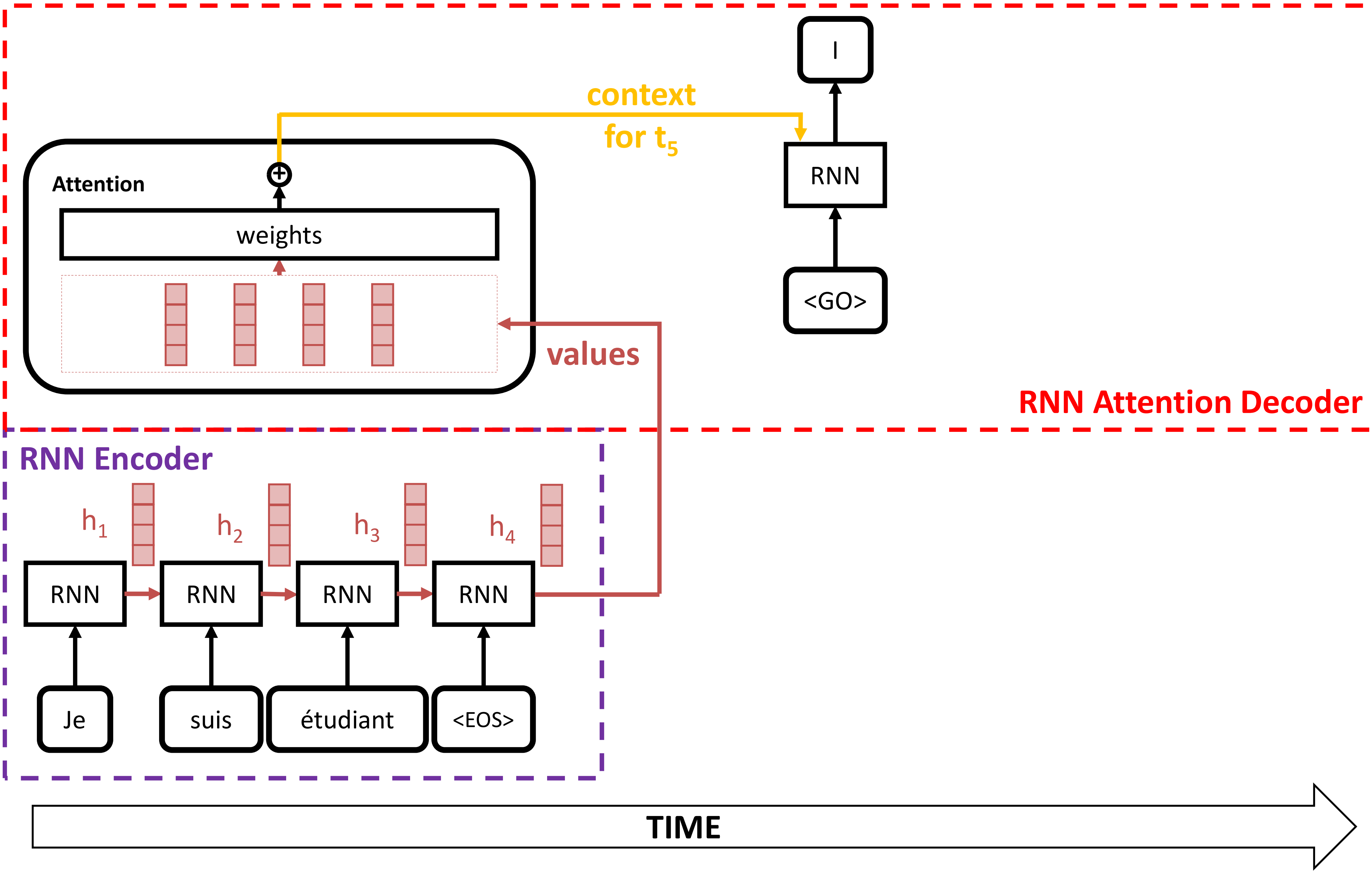
RNN Encoder-Decoder with Attention



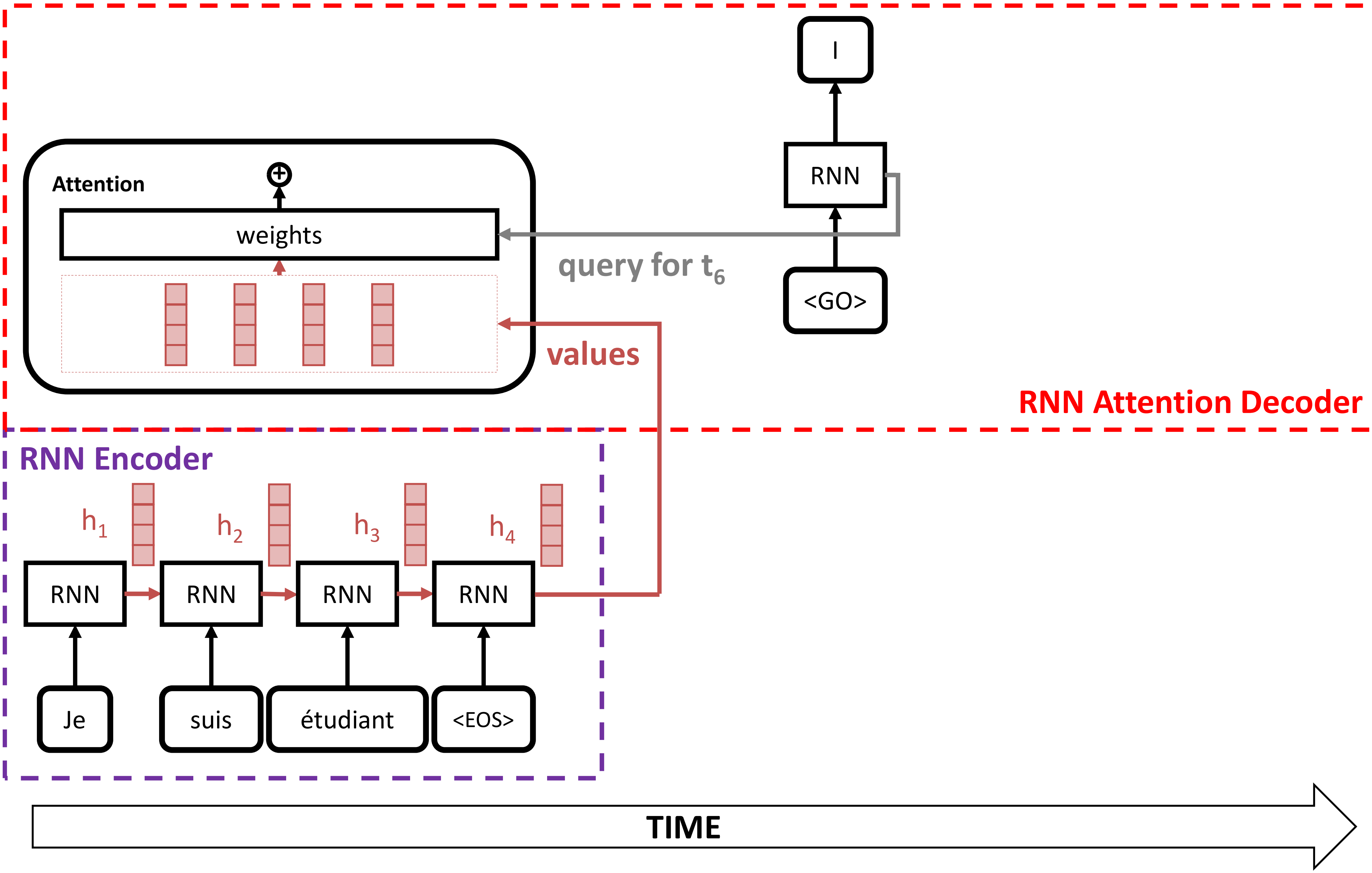
RNN Encoder-Decoder with Attention



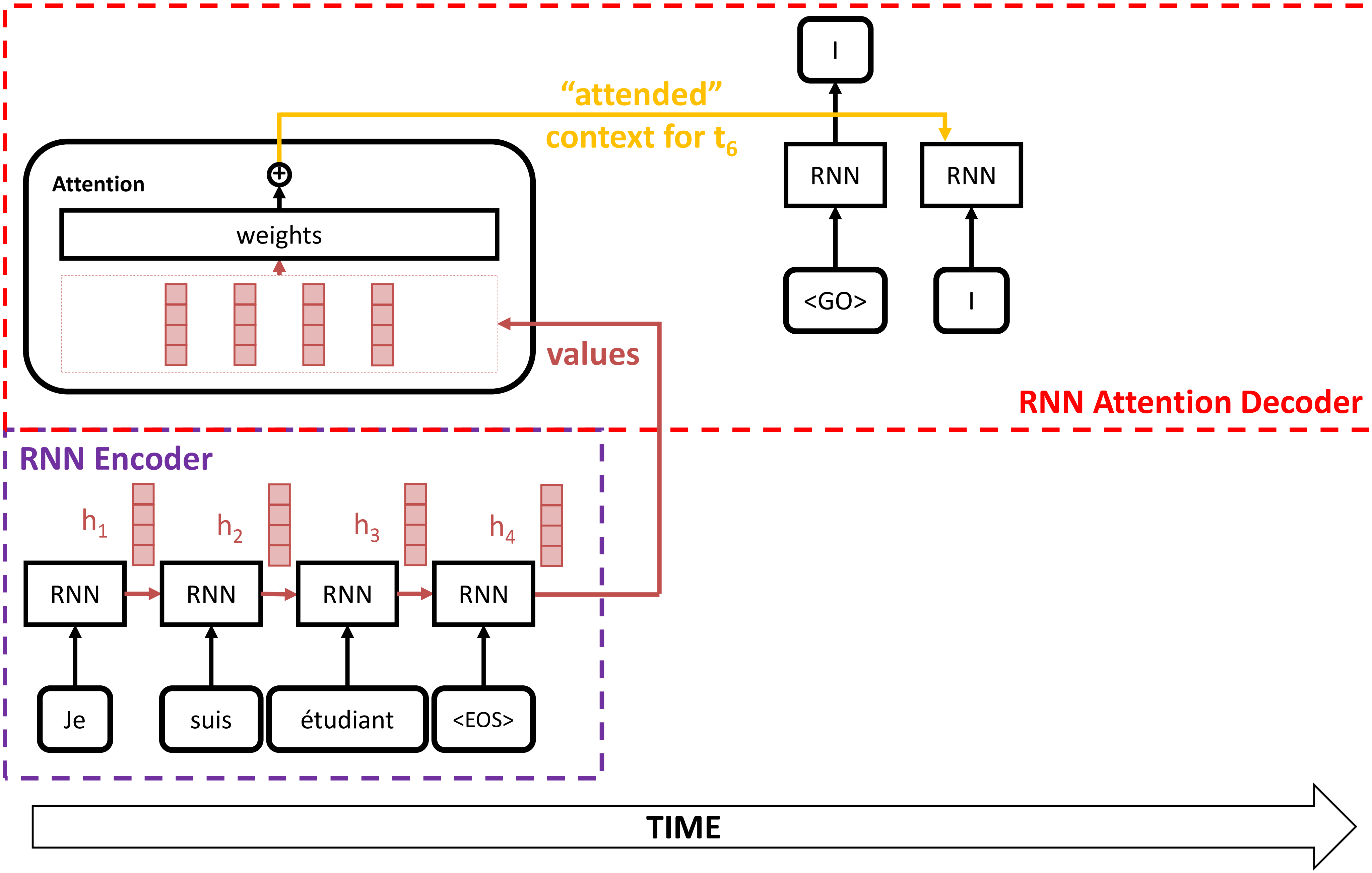
RNN Encoder-Decoder with Attention



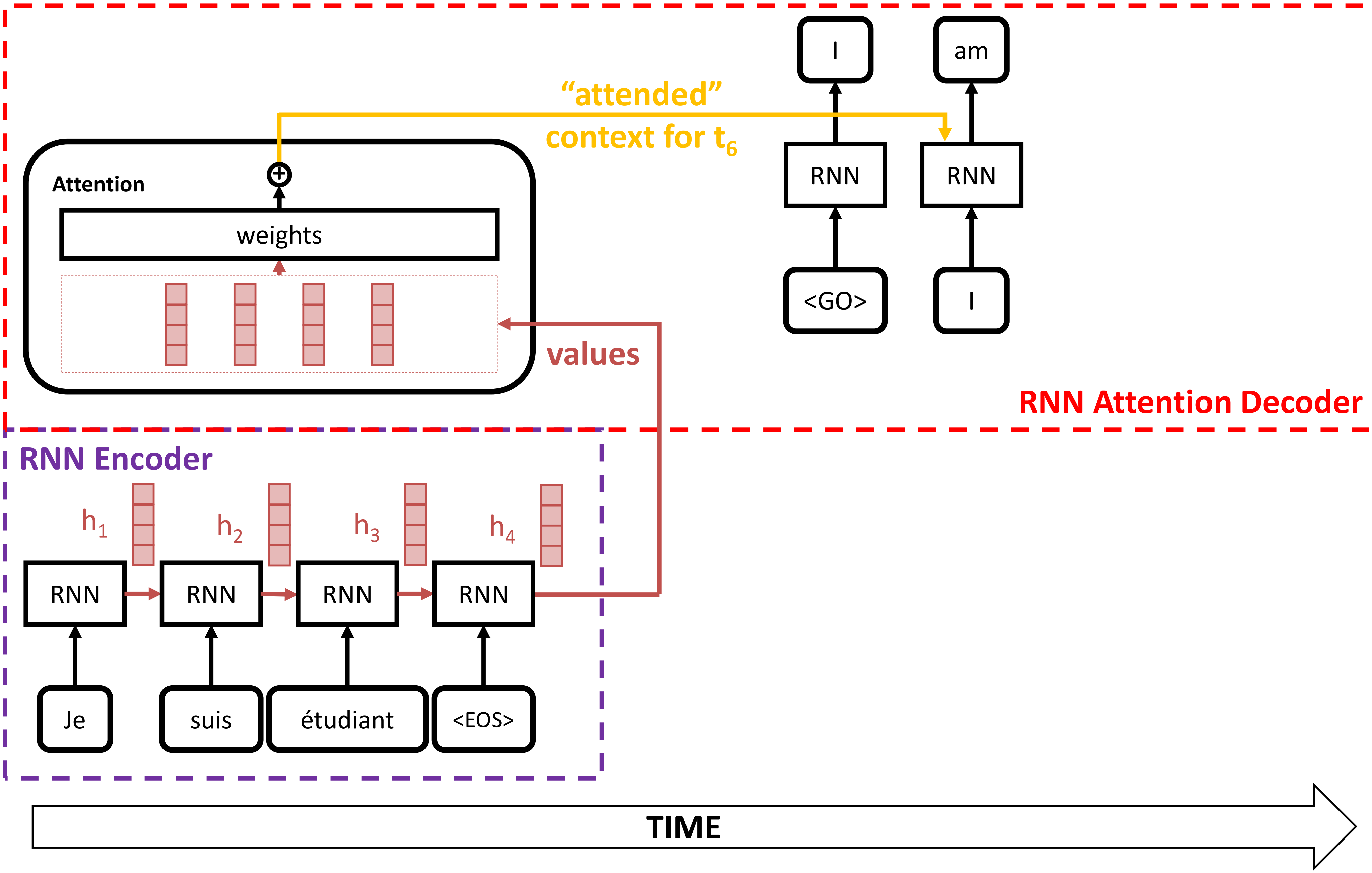
RNN Encoder-Decoder with Attention



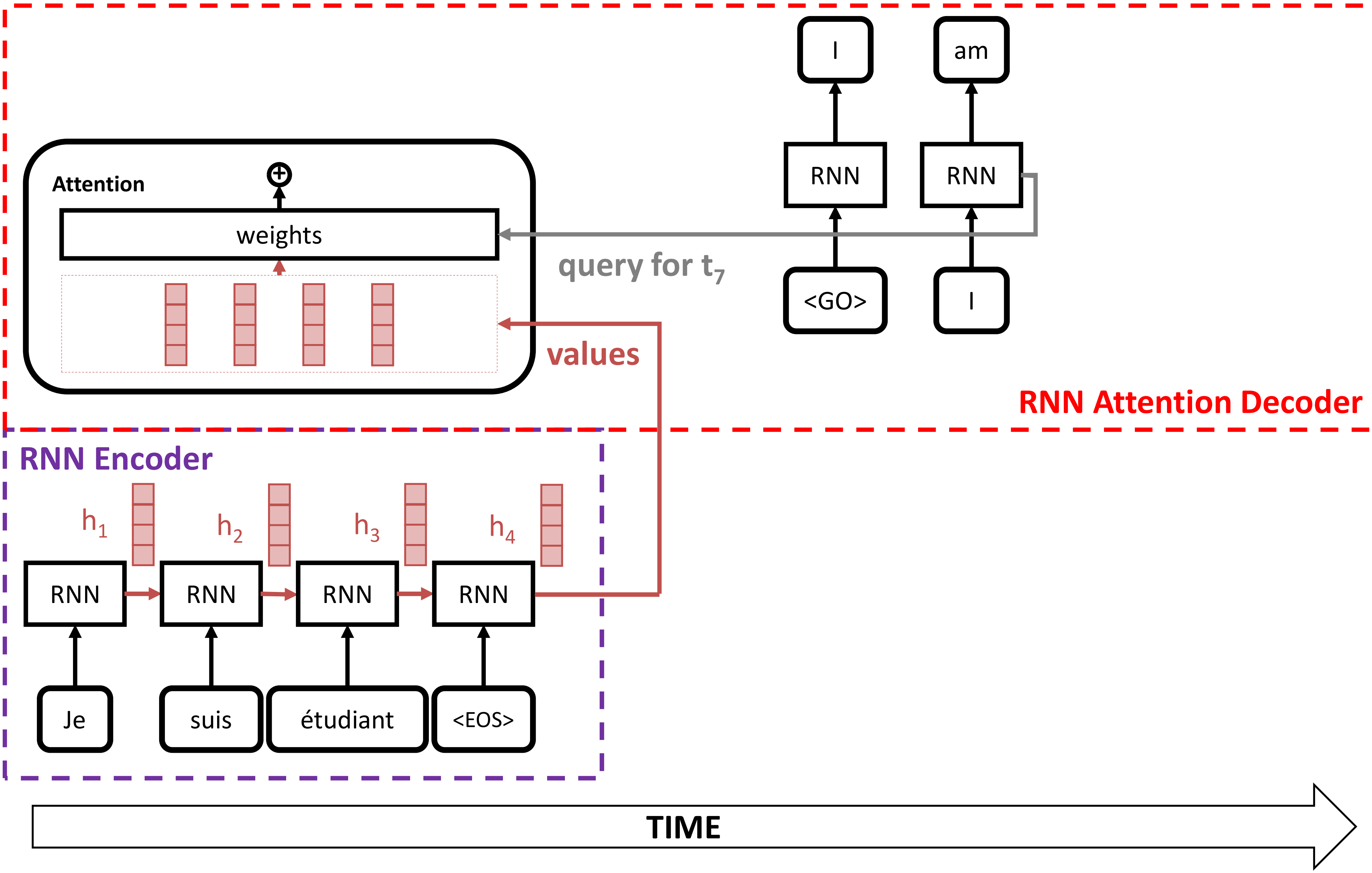
RNN Encoder-Decoder with Attention



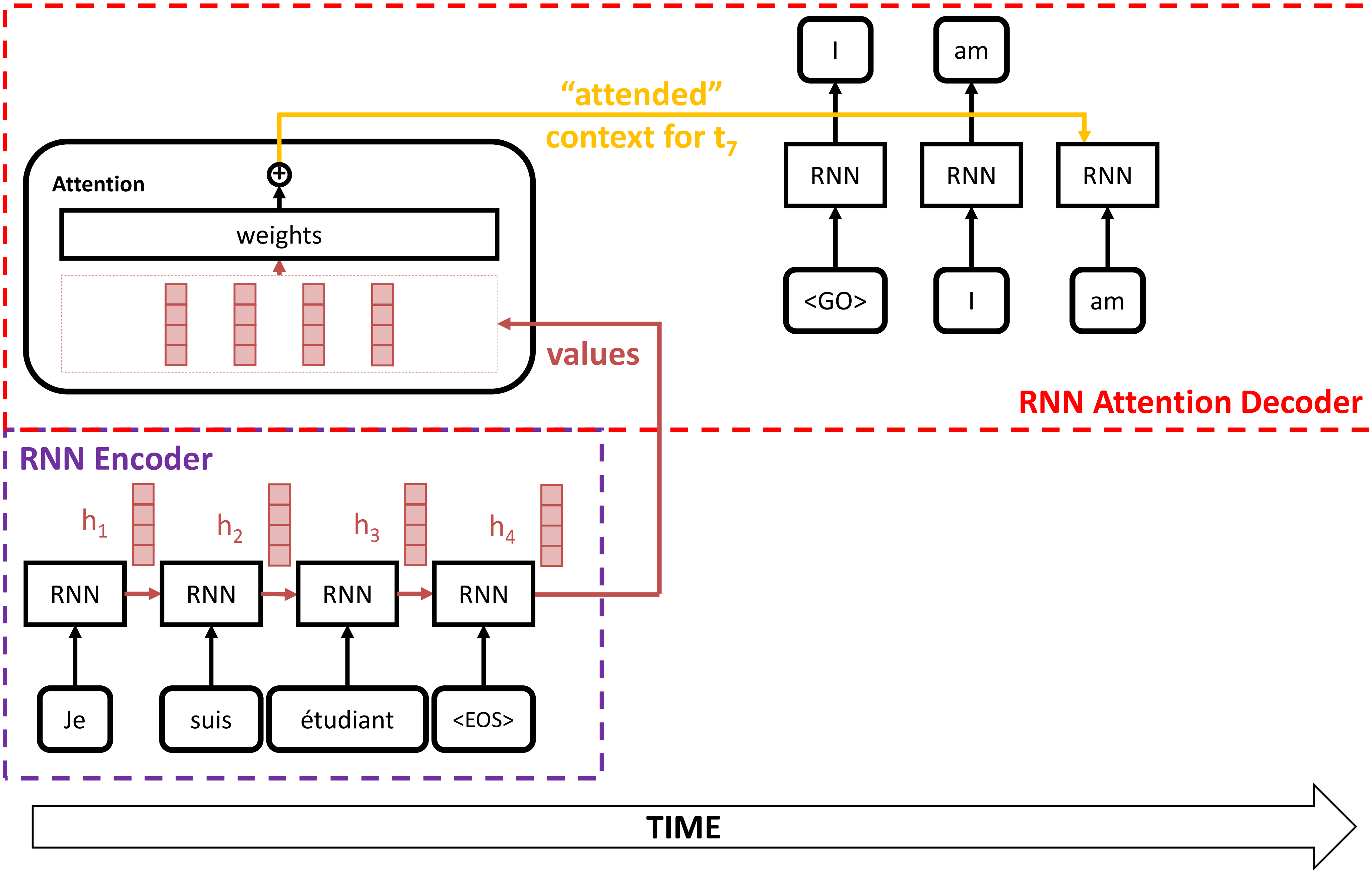
RNN Encoder-Decoder with Attention



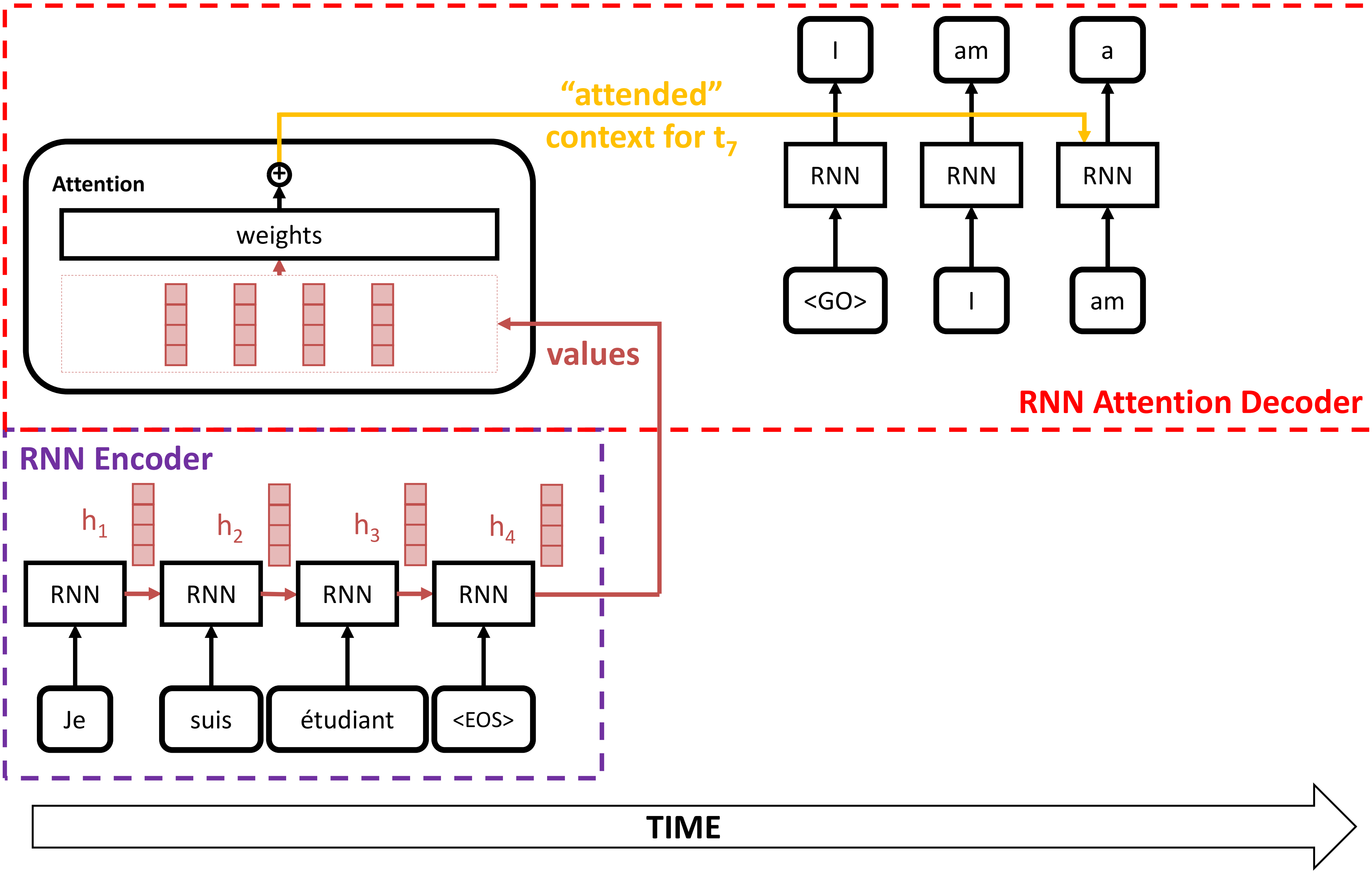
RNN Encoder-Decoder with Attention



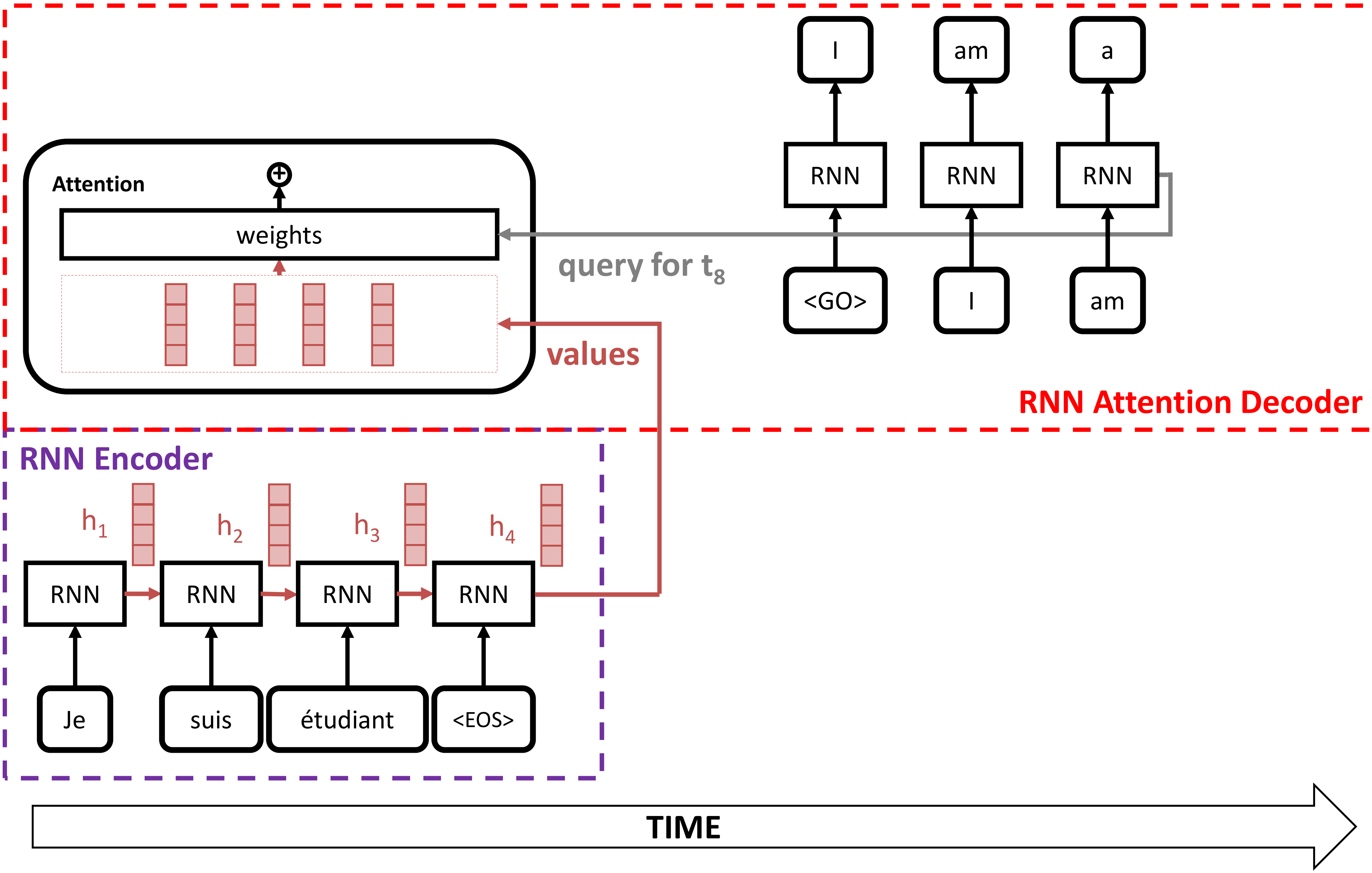
RNN Encoder-Decoder with Attention



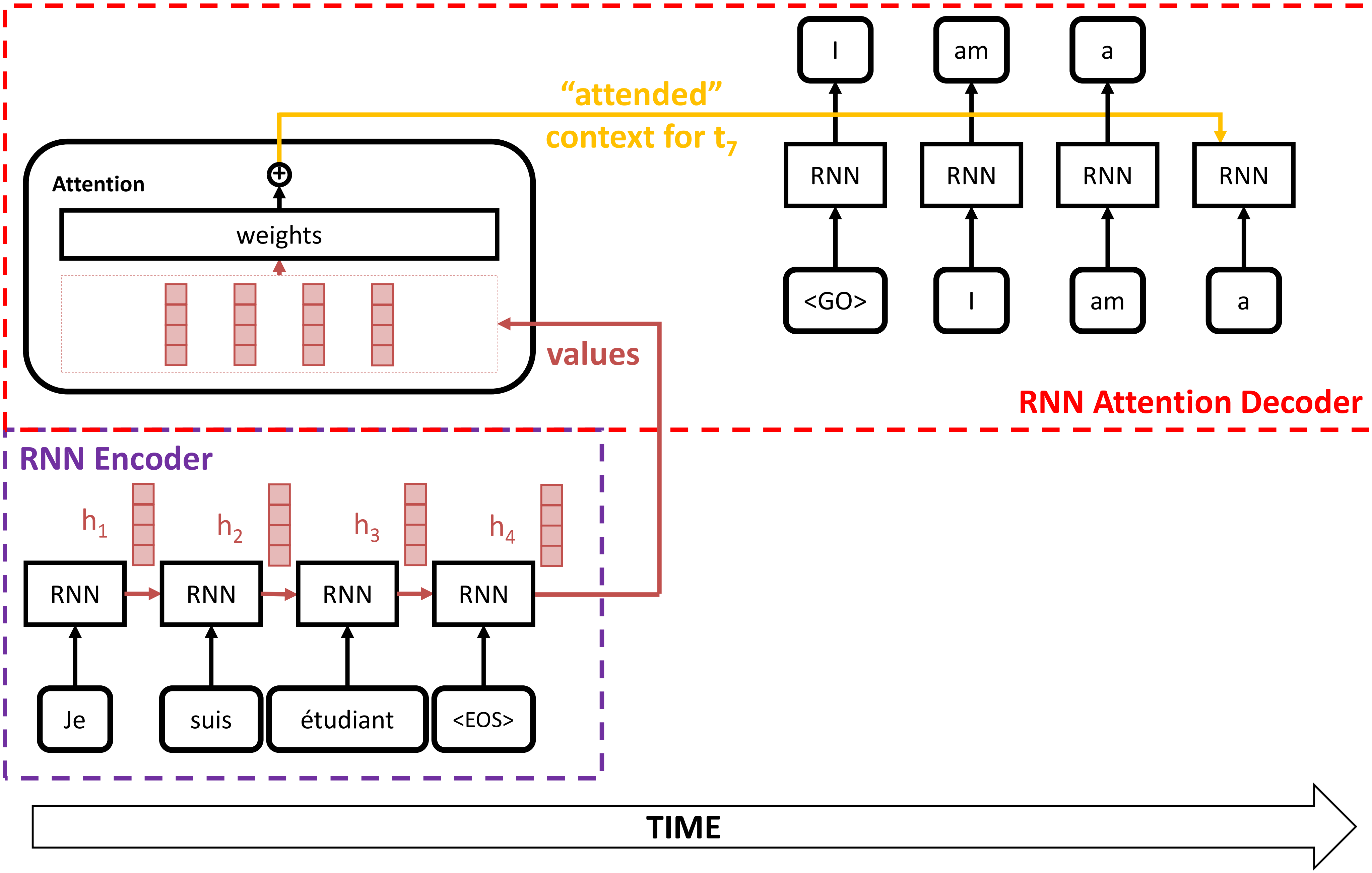
RNN Encoder-Decoder with Attention



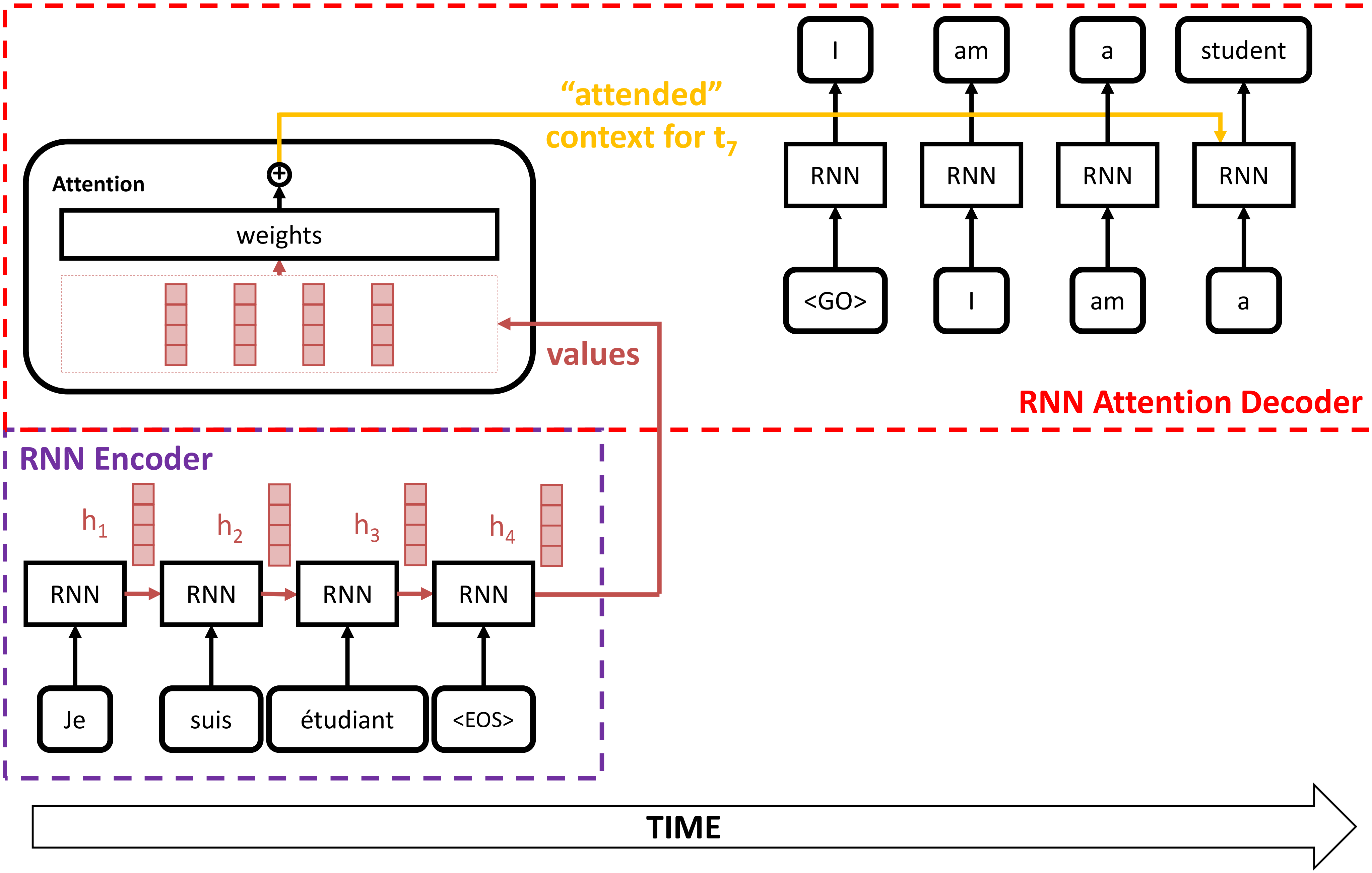
RNN Encoder-Decoder with Attention



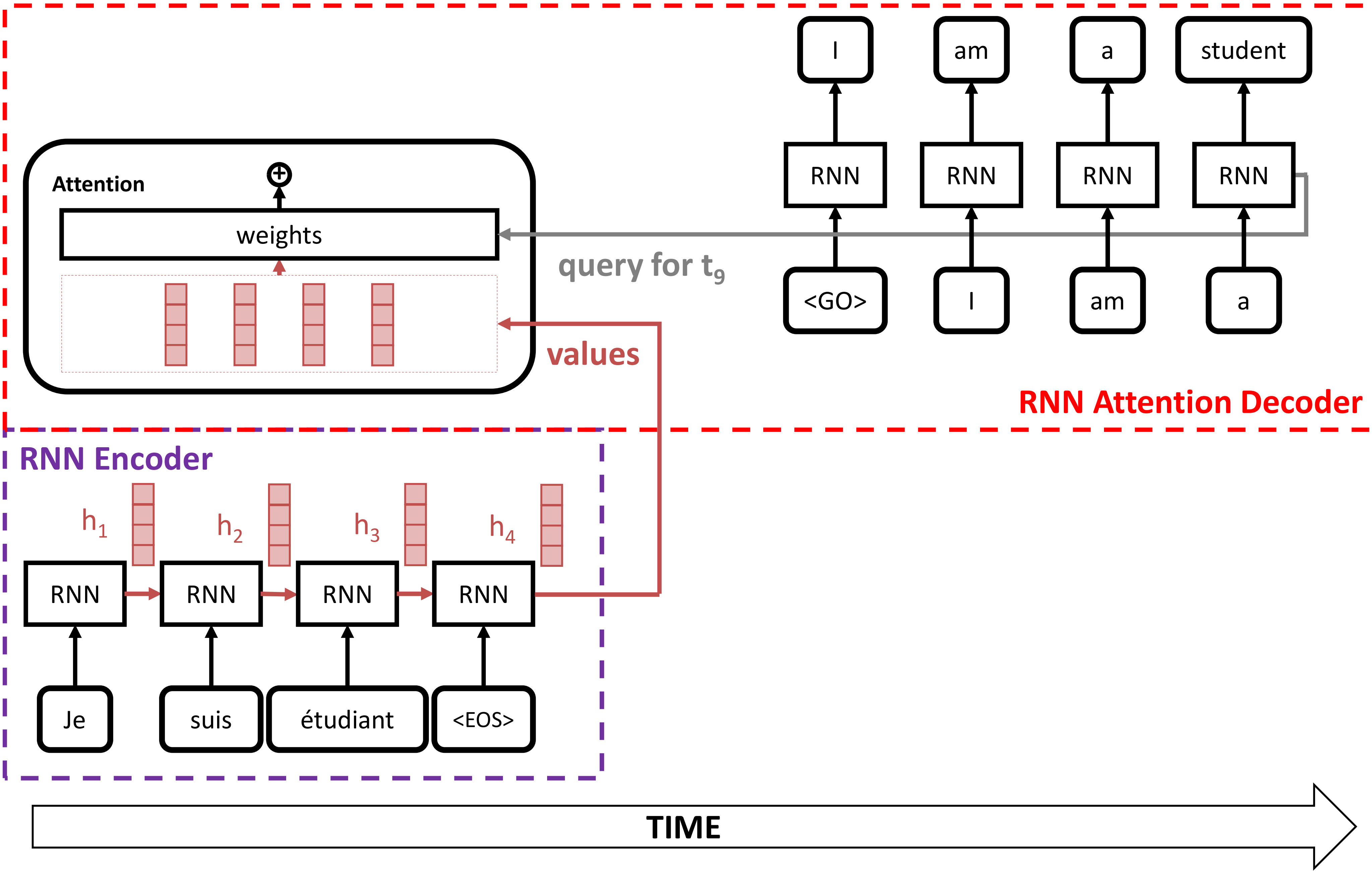
RNN Encoder-Decoder with Attention



RNN Encoder-Decoder with Attention

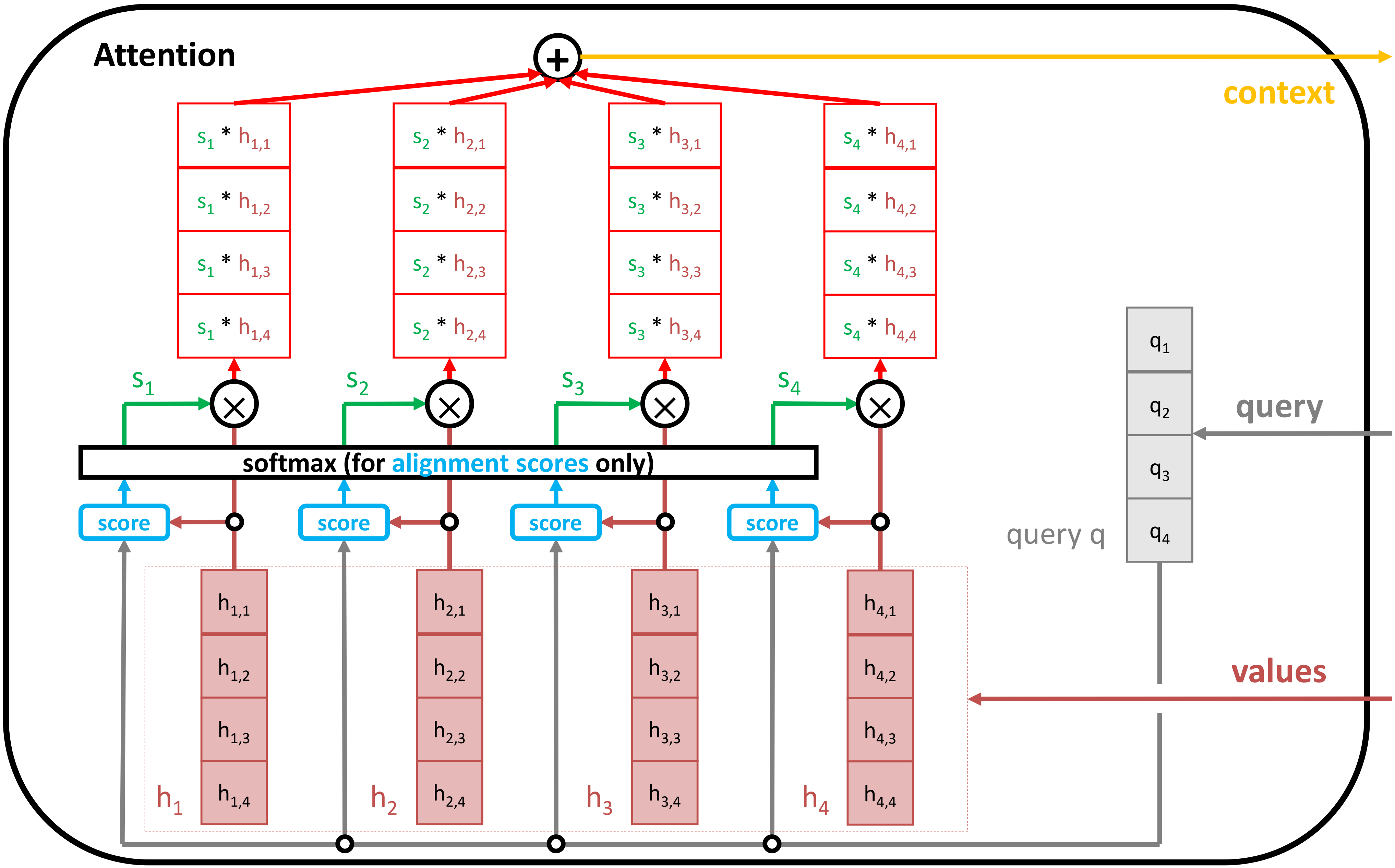


RNN Encoder-Decoder with Attention

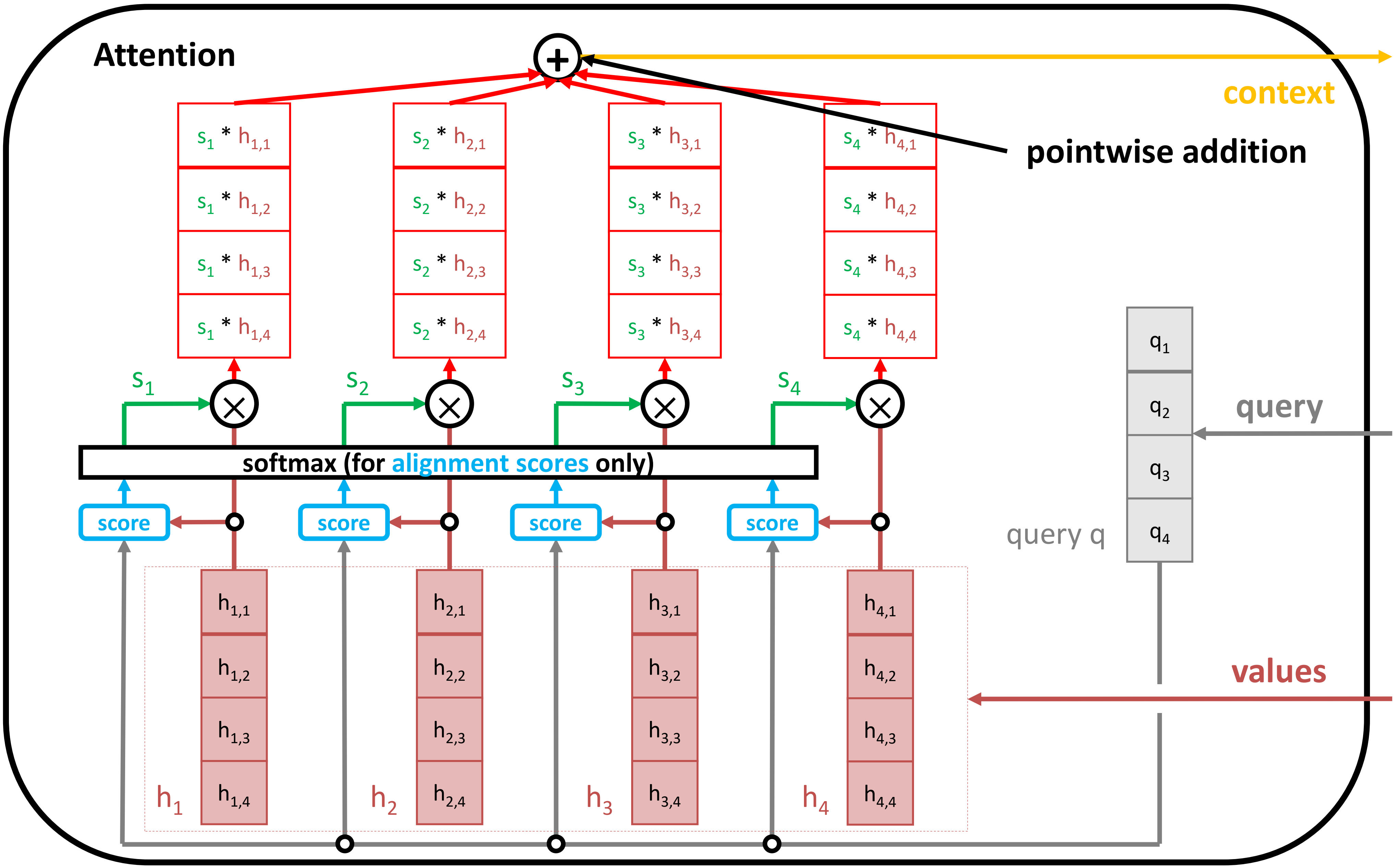


What is inside the RNN Decoder Attention?

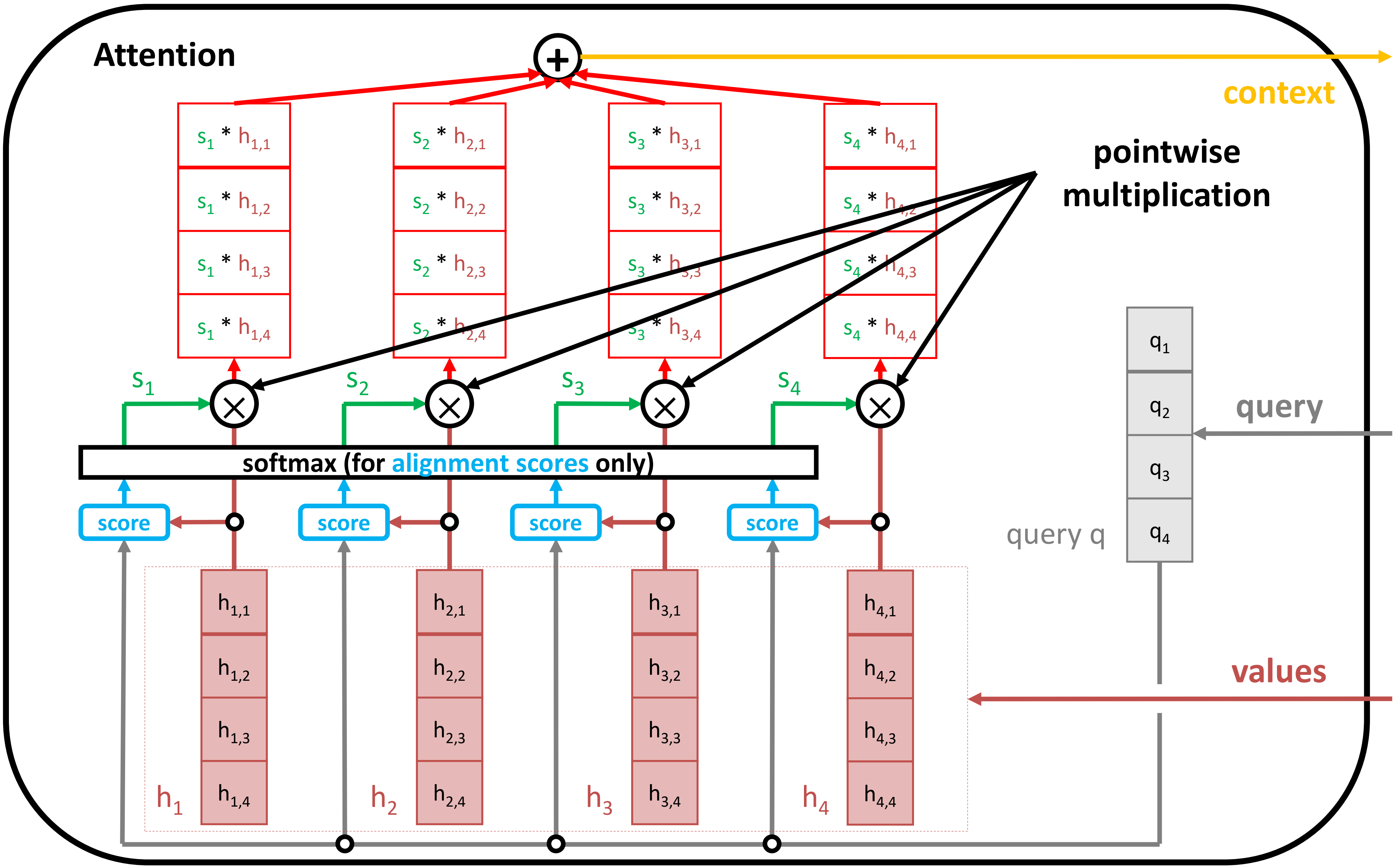
RNN Encoder-Decoder Attention



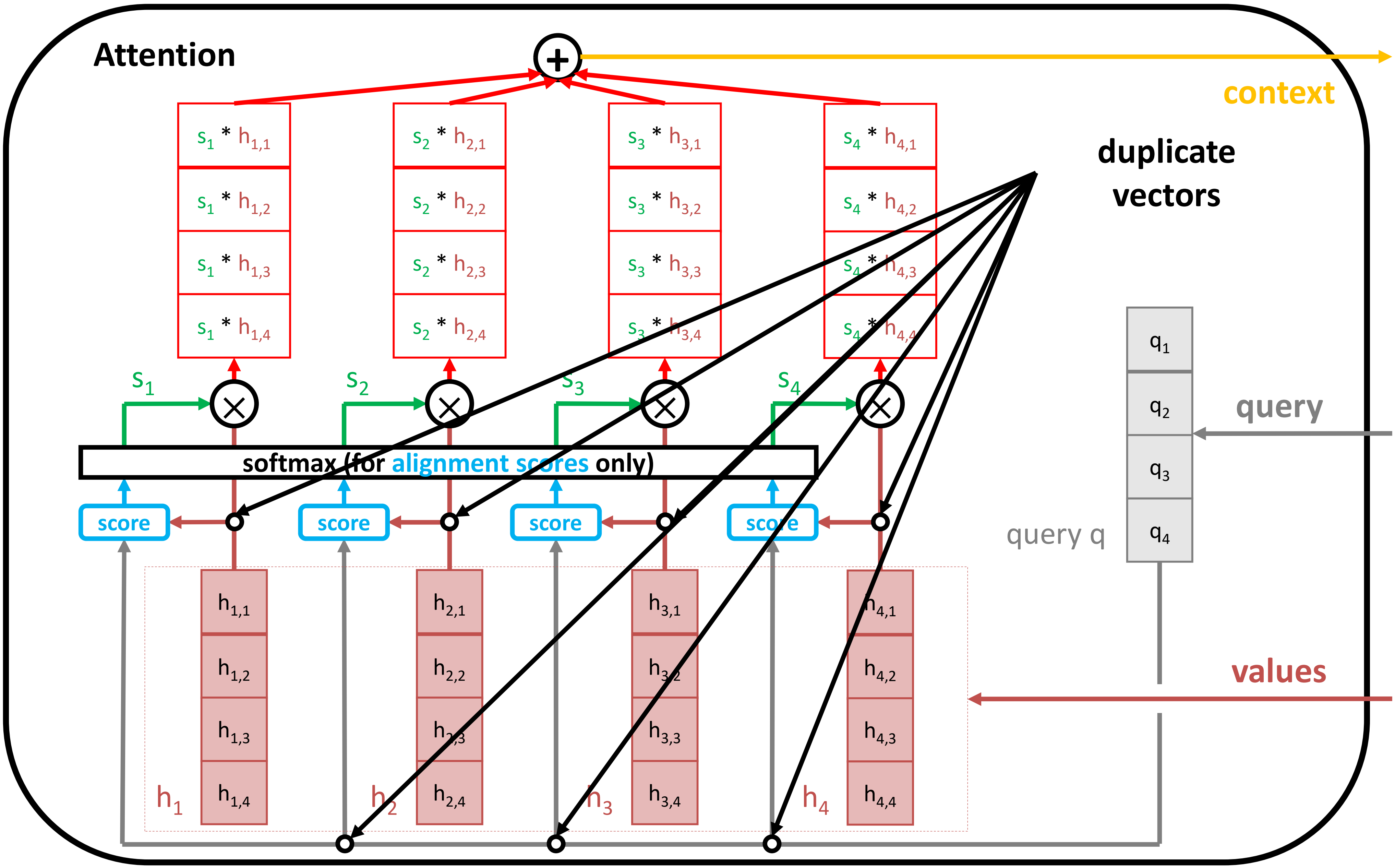
RNN Encoder-Decoder Attention



RNN Encoder-Decoder Attention



RNN Encoder-Decoder Attention



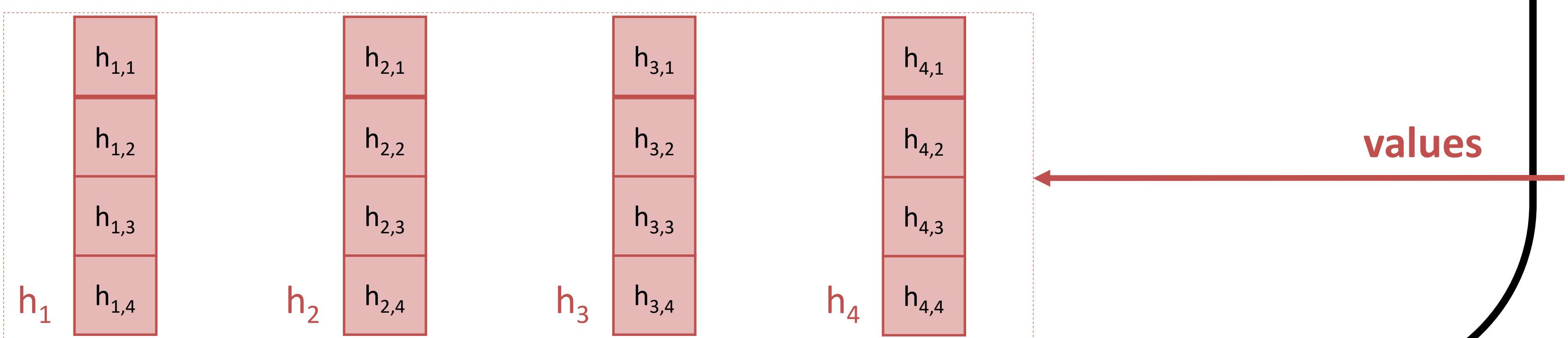
What is RNN Decoder Attention doing?

RNN Encoder-Decoder Attention

Attention

1. **Prepare inputs**

a) encoder hidden states



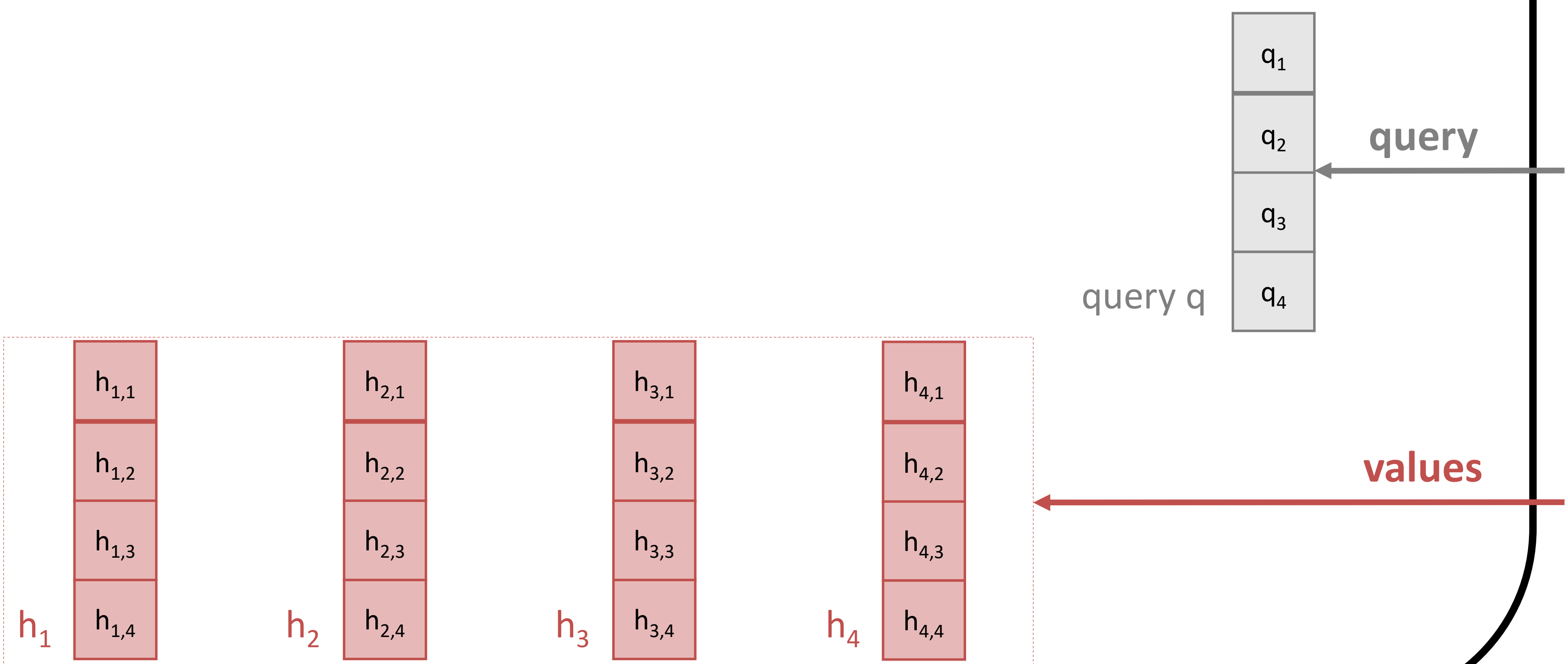
RNN Encoder-Decoder Attention

Attention

1. **Prepare inputs**

a) **encoder hidden states**

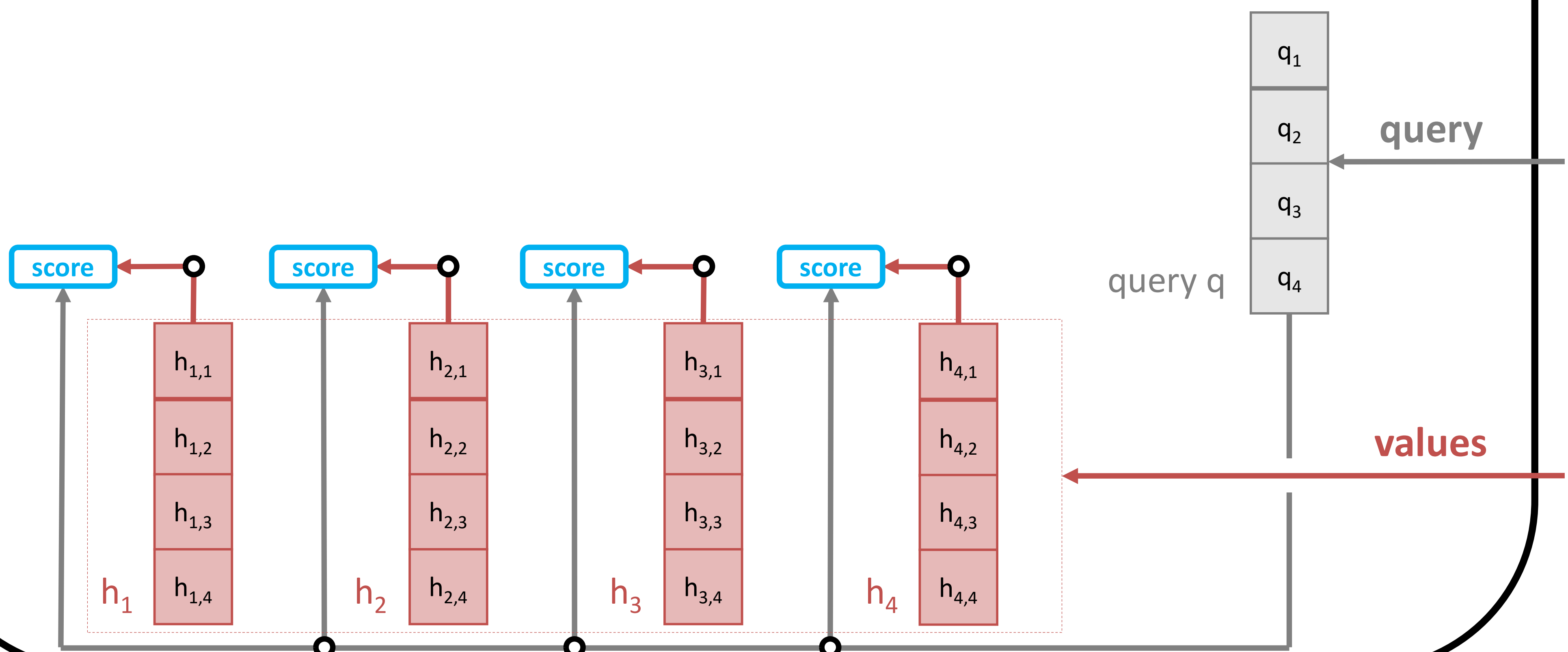
b) **query for (next) time t_5**



RNN Encoder-Decoder Attention

Attention

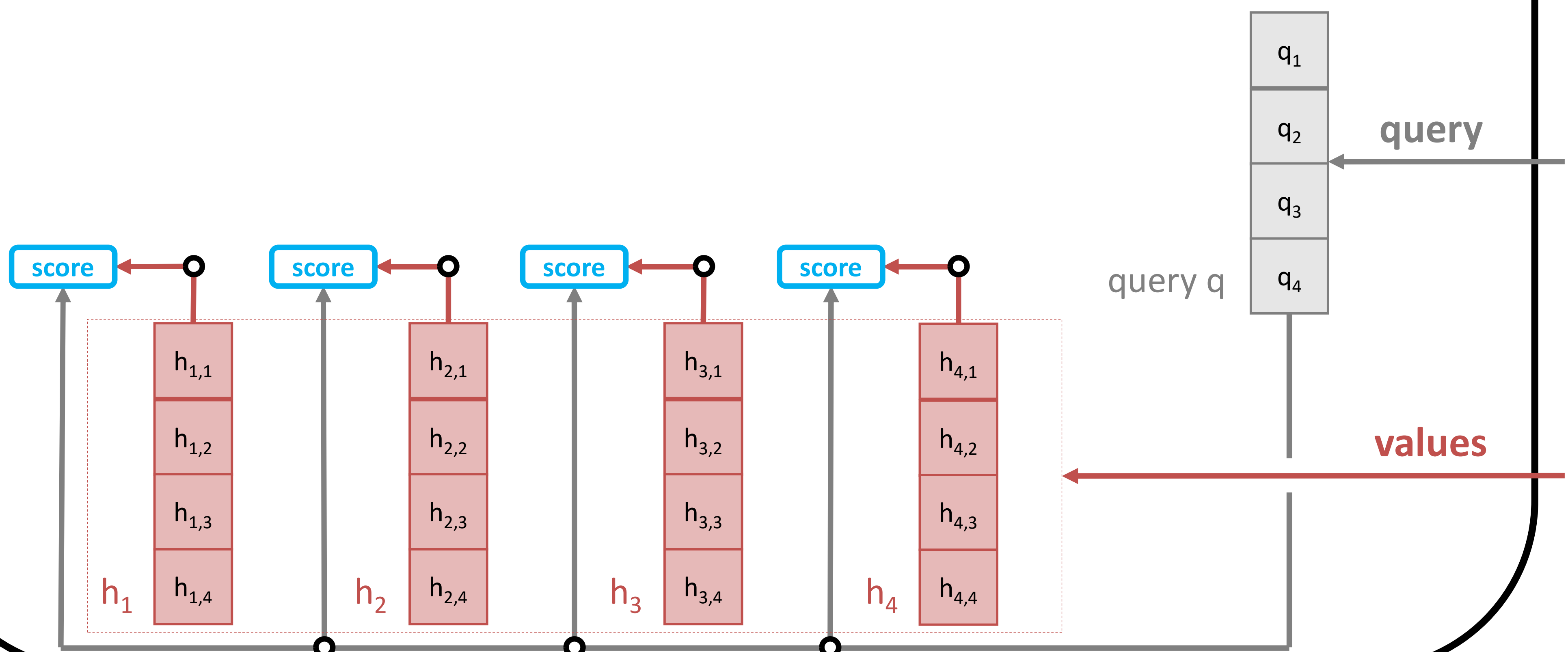
2. Score hidden states
- measure similarity
between q and h_i



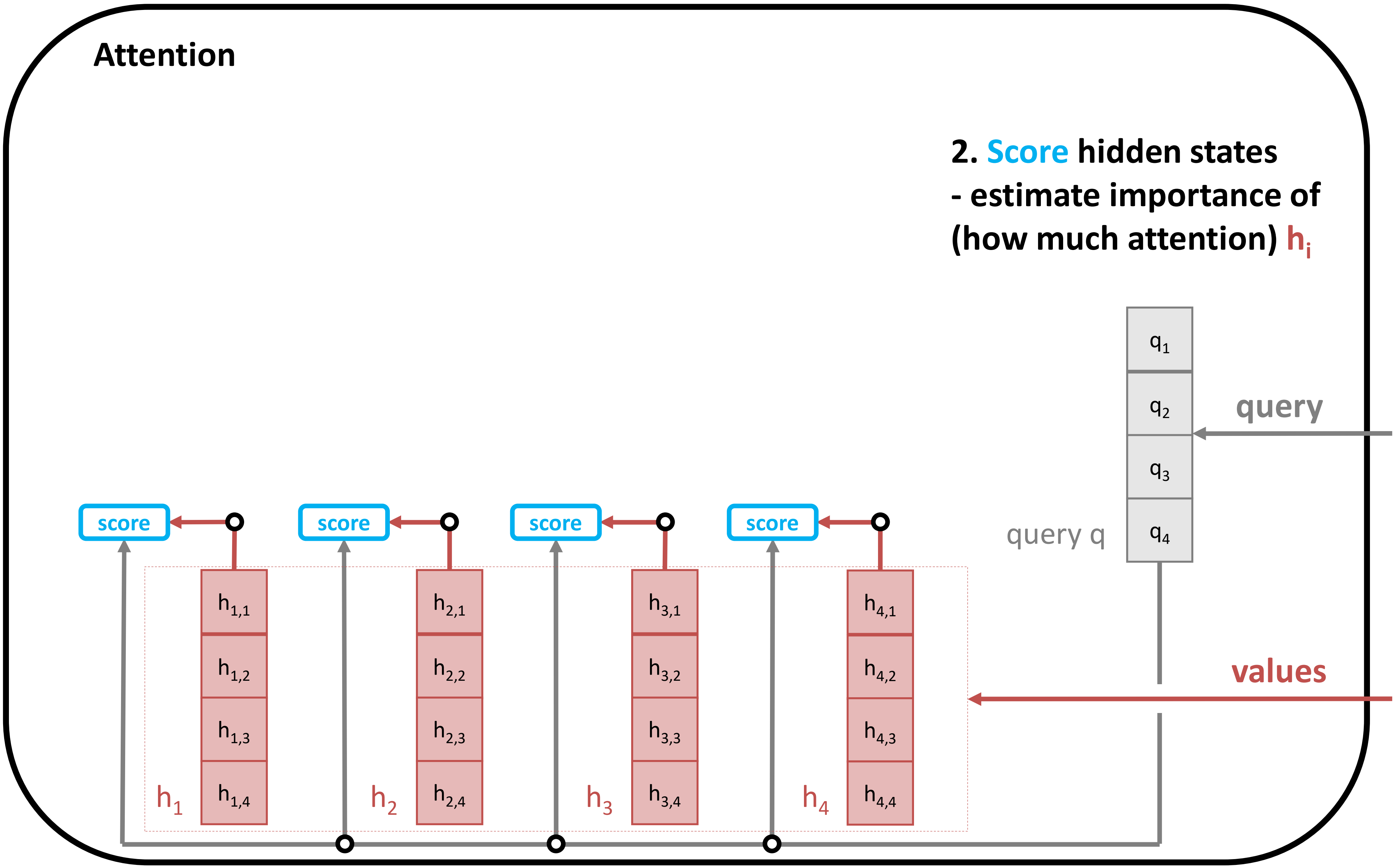
RNN Encoder-Decoder Attention

Attention

2. Score hidden states
- measure alignment between q and h_i



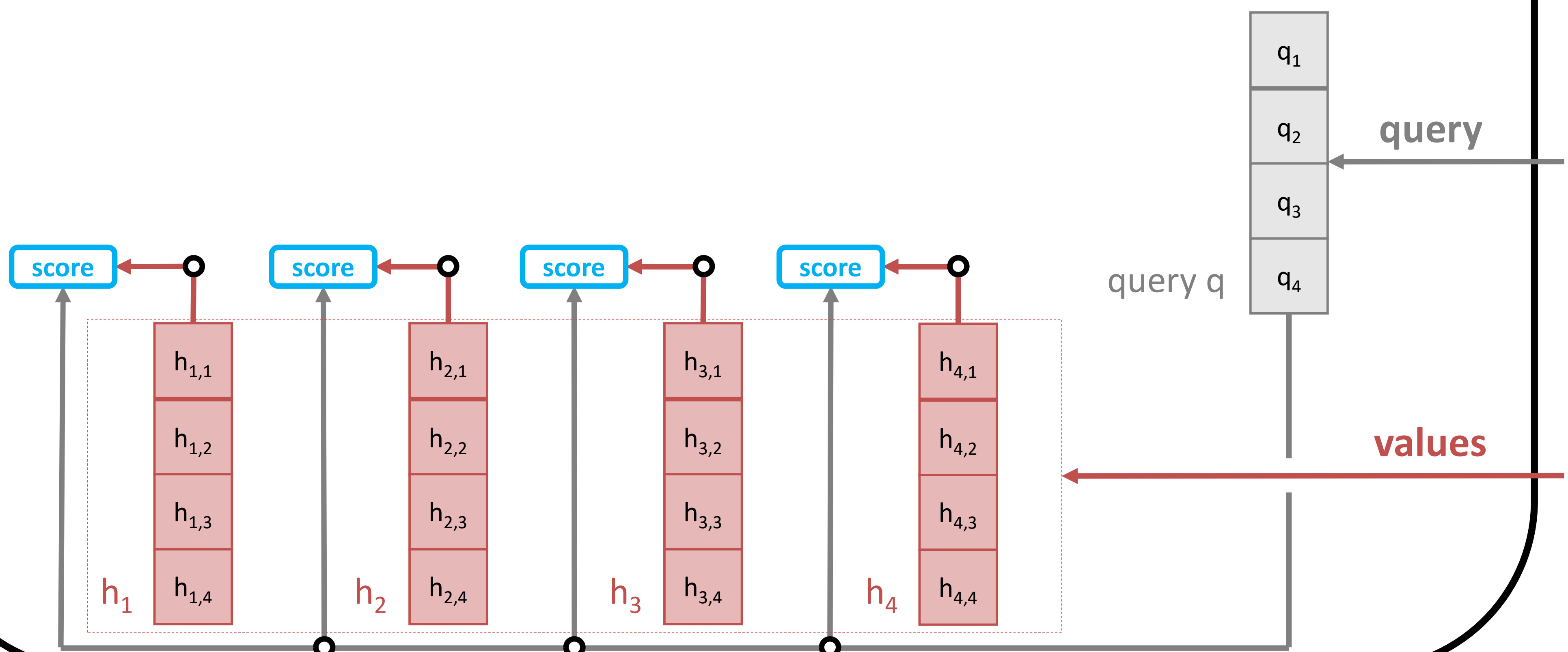
RNN Encoder-Decoder Attention



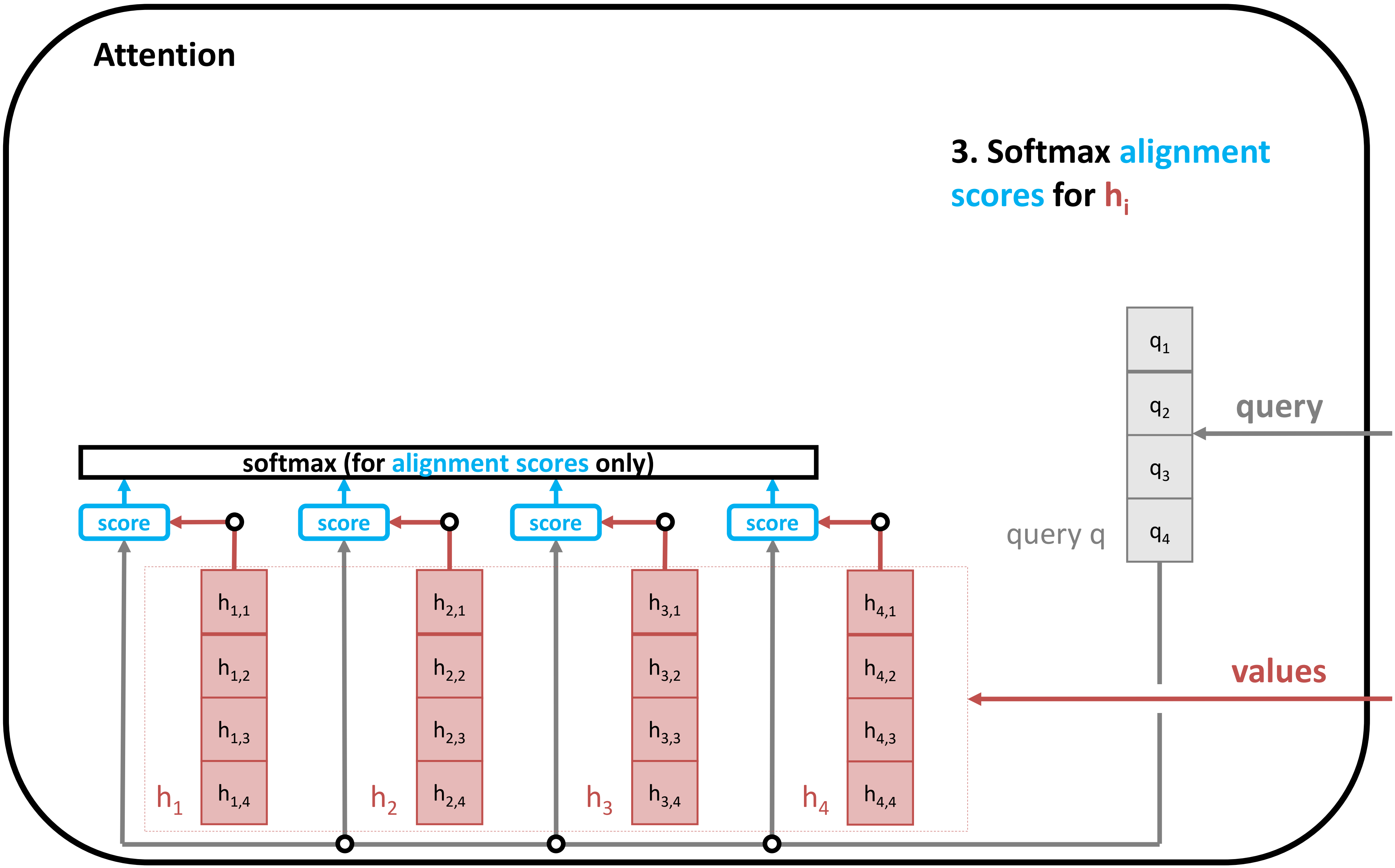
RNN Encoder-Decoder Attention

Attention

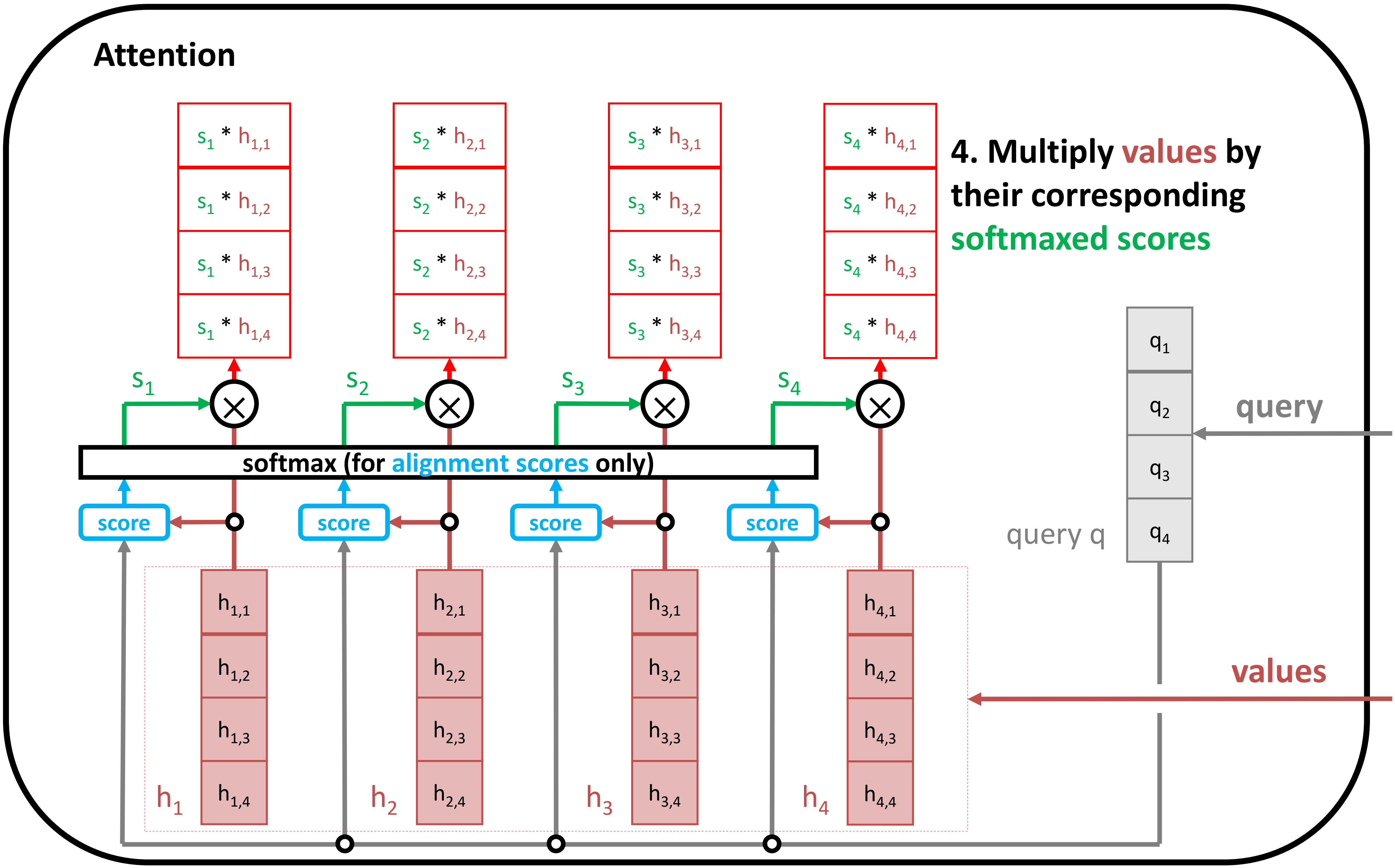
2. Score hidden states - calculate attention weights for h_i



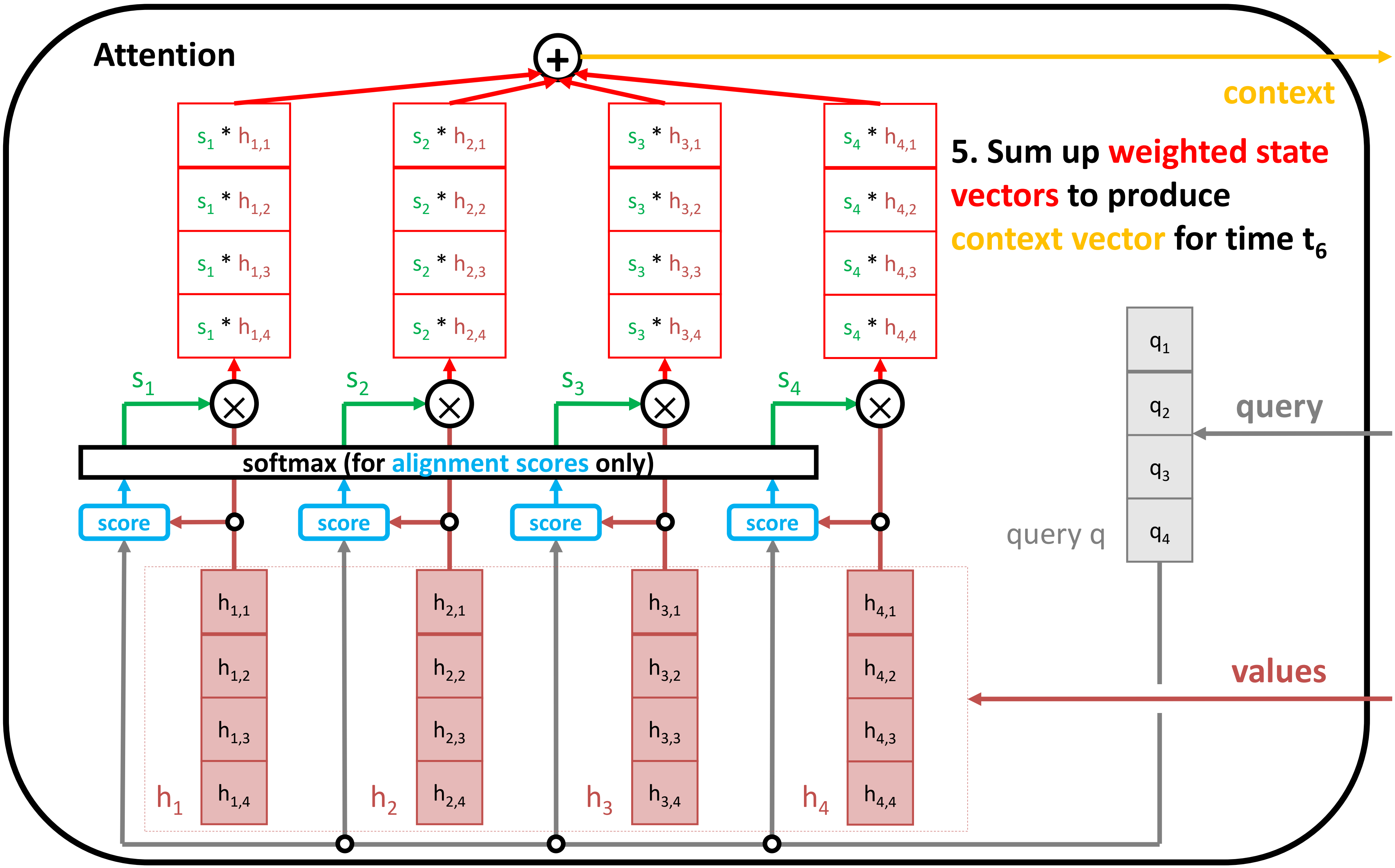
RNN Encoder-Decoder Attention



RNN Encoder-Decoder Attention



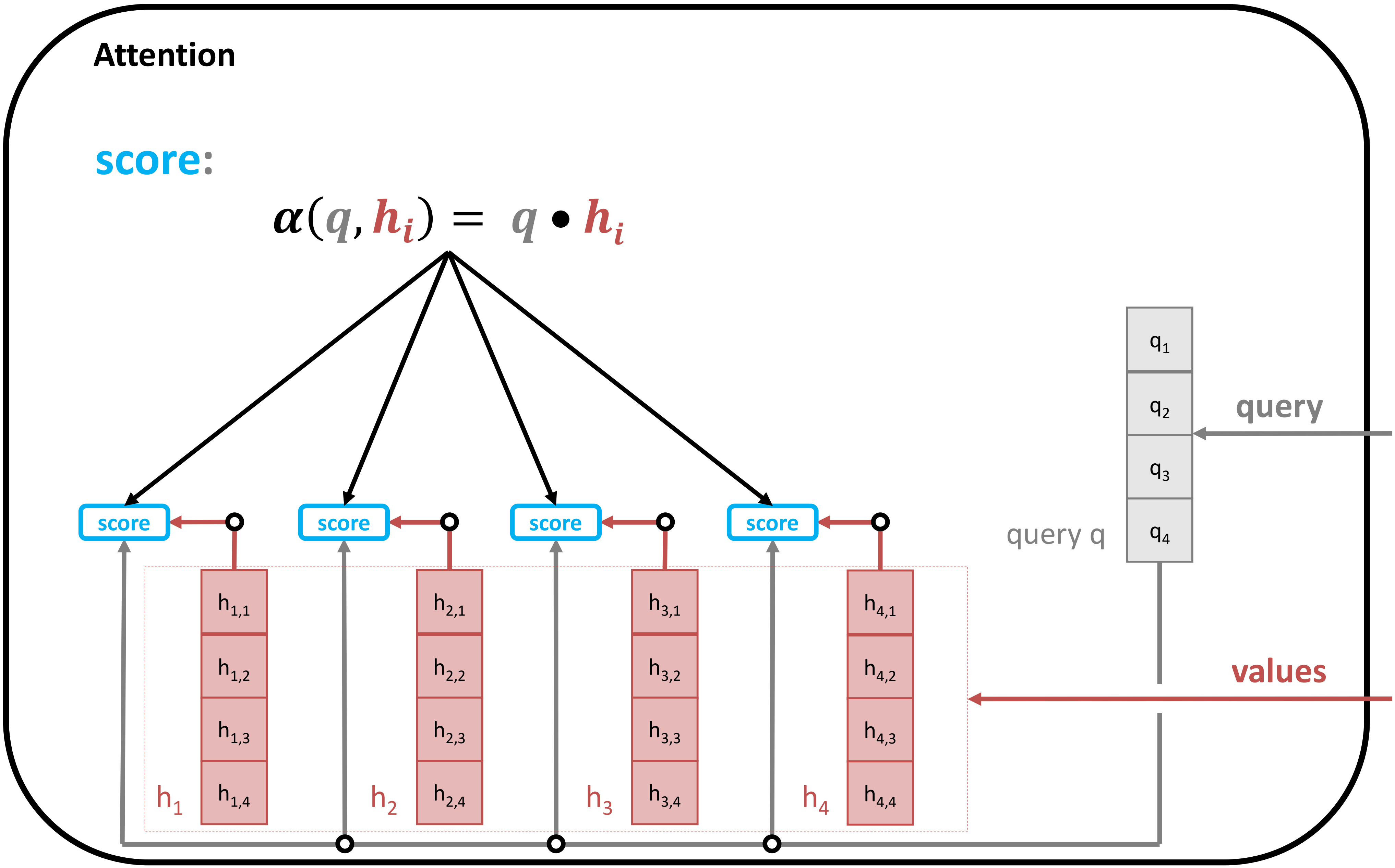
RNN Encoder-Decoder Attention



Summary / Intuition

- Attention mechanism creates context dynamically
- “Attended” values “dominate” context
  - this allows the decoder to dynamically focus on most relevant aspects of the encoder output
- The query represents/summarizes the “current task” of the decoder
  - the input it is processing within the current context
- Scoring for values estimates similarity (between a query and specific value) values:
  - dot product
  - scaled dot product
  - etc.

Score: Dot Product

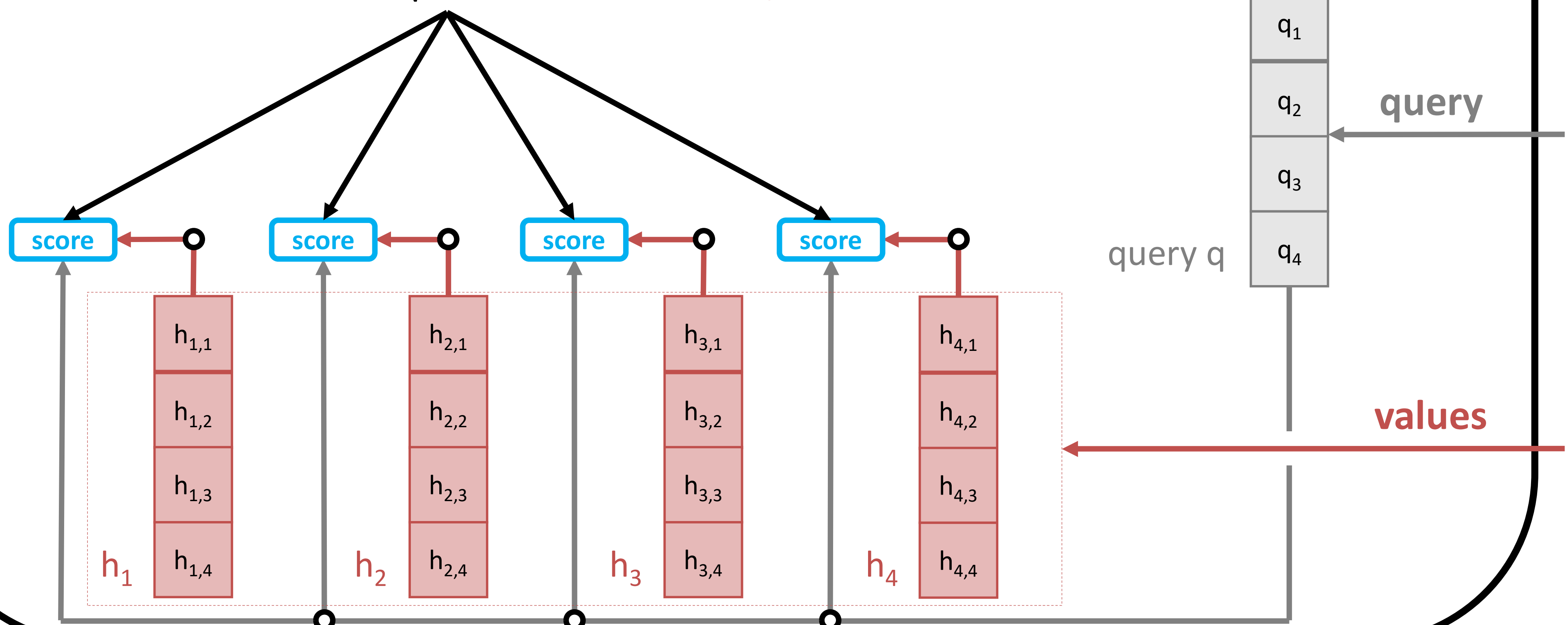


Score: Scaled Dot Product

Attention

score:

$$\alpha(q, h_i) = \frac{q \bullet h_i}{\sqrt{dimension_{h_i}}}$$

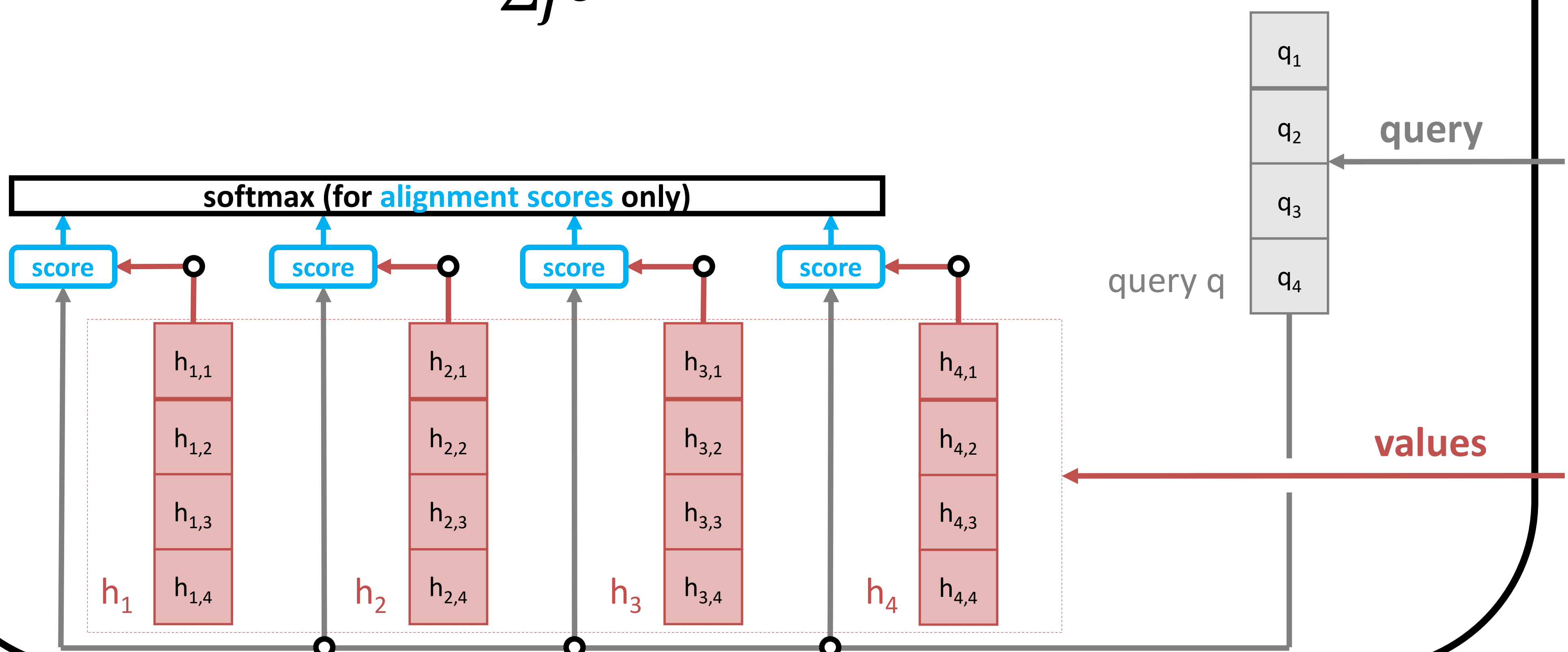


Softmax

Attention

softmax on scores:

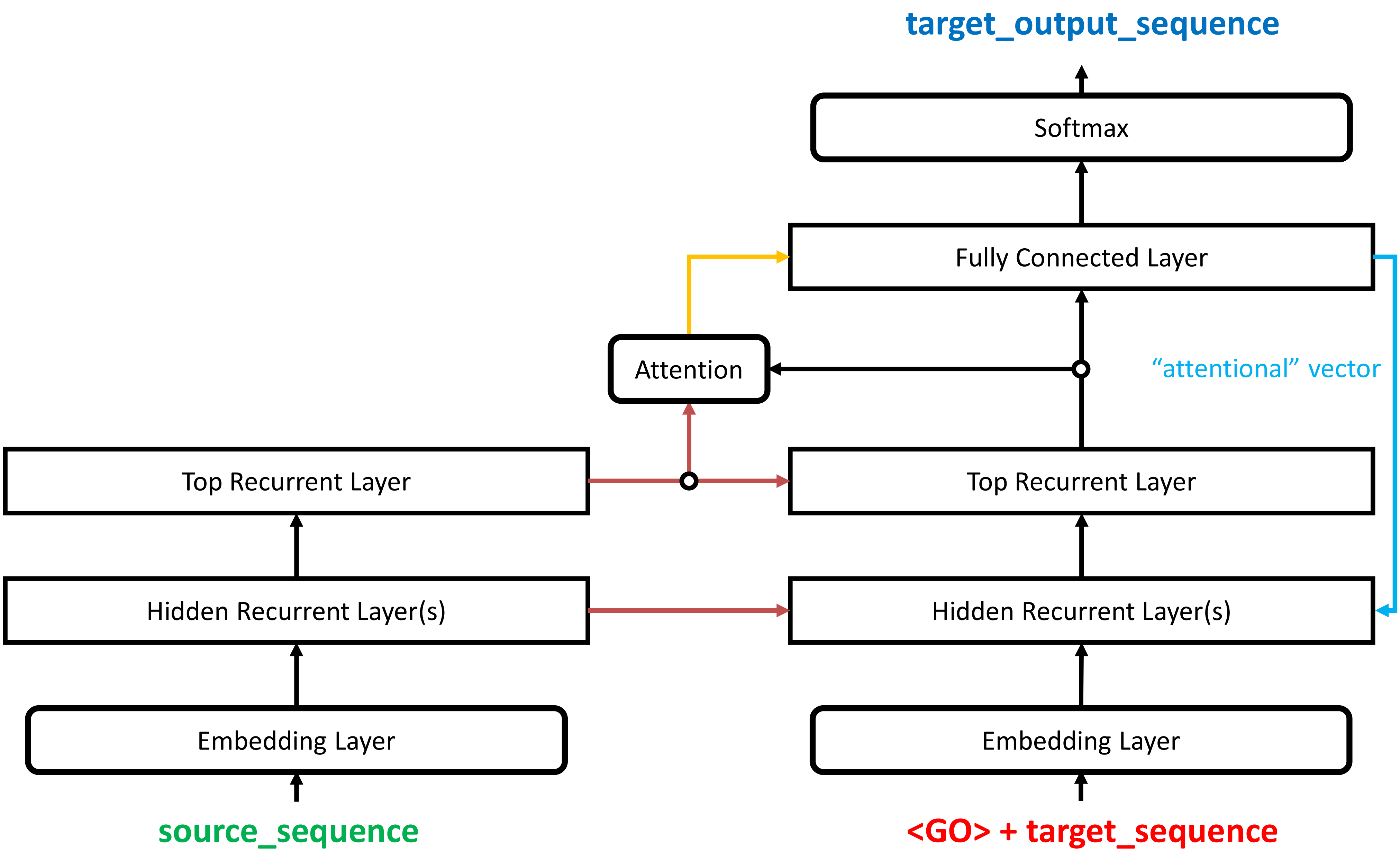
$$s(score_i) = \frac{e^{score_i}}{\sum_j e^{score_j}}$$



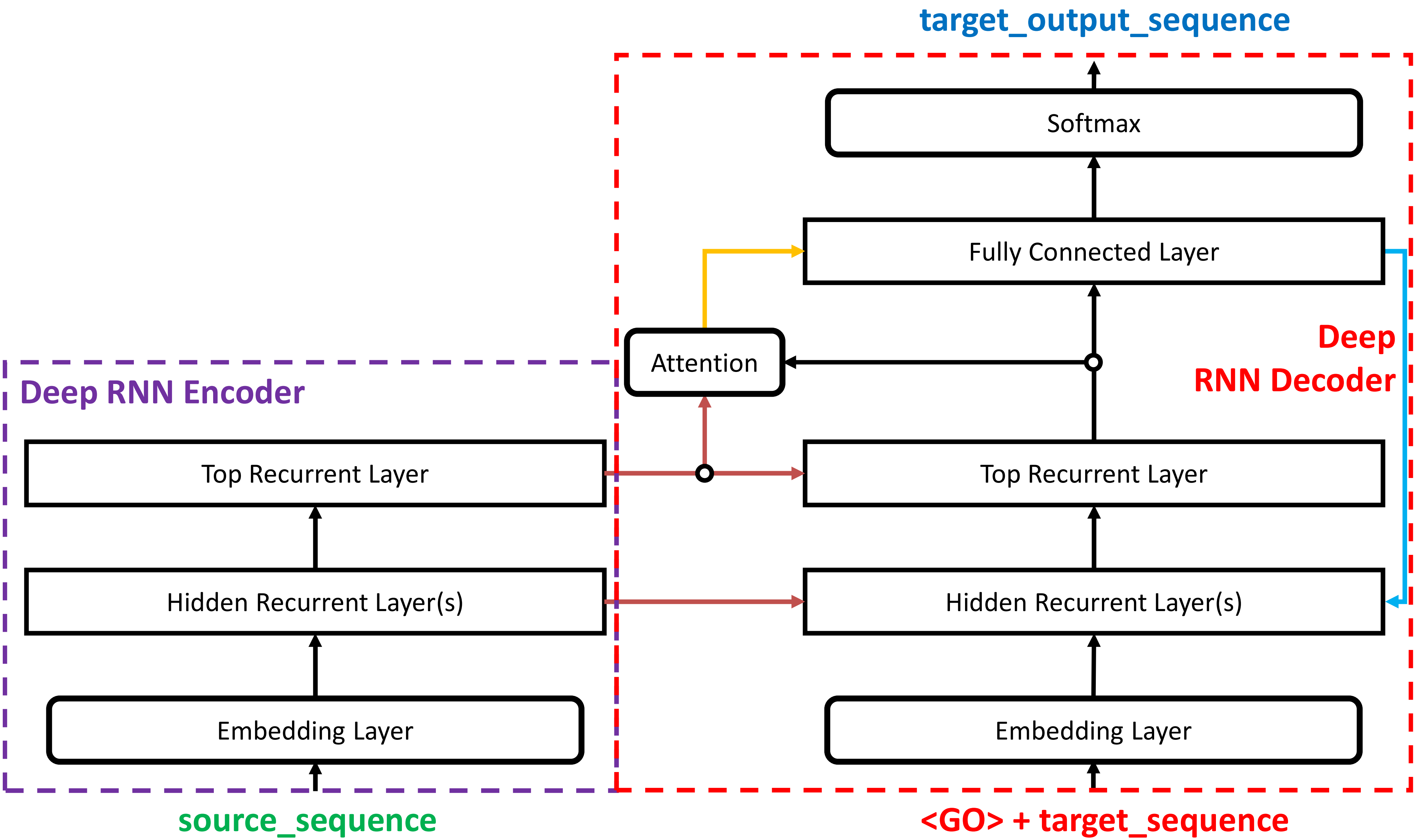
Benefits of the Attention Structure

- Significantly improves performance (in many applications)
  - it's very useful to allow the decoder to focus on certain parts of the source
- Solves the bottleneck issue
  - attention allows decoder to look directly at the source (and "bypass" the bottleneck)
- Helps with vanishing gradient problem
  - provides shortcut to far away states
- Provides some interpretability
  - inspecting attention distribution we can see what the decoder was focusing on

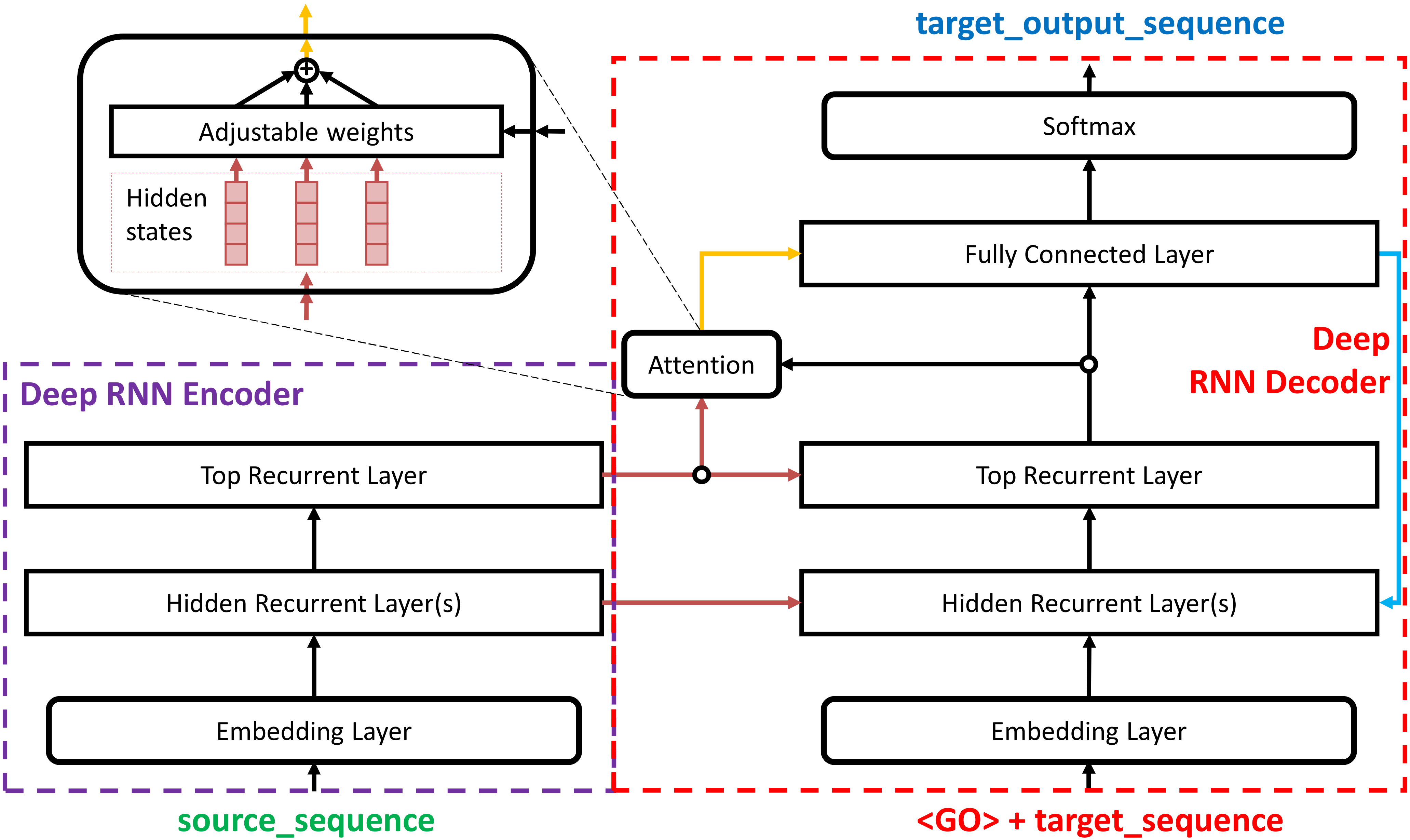
Deep RNN Enc-Dec with Attention



Deep RNN Enc-Dec with Attention



Deep RNN Enc-Dec with Attention



Transformers - Basics

Generative Pre-trained Transformer 3

What is it?

Generative Pre-trained Transformer 3 (GPT-3) is an **autoregressive language model that uses deep learning to produce human-like text**. It is the third-generation language prediction model in the GPT-n series (and the successor to GPT-2) created by OpenAI, a San Francisco-based artificial intelligence research laboratory.

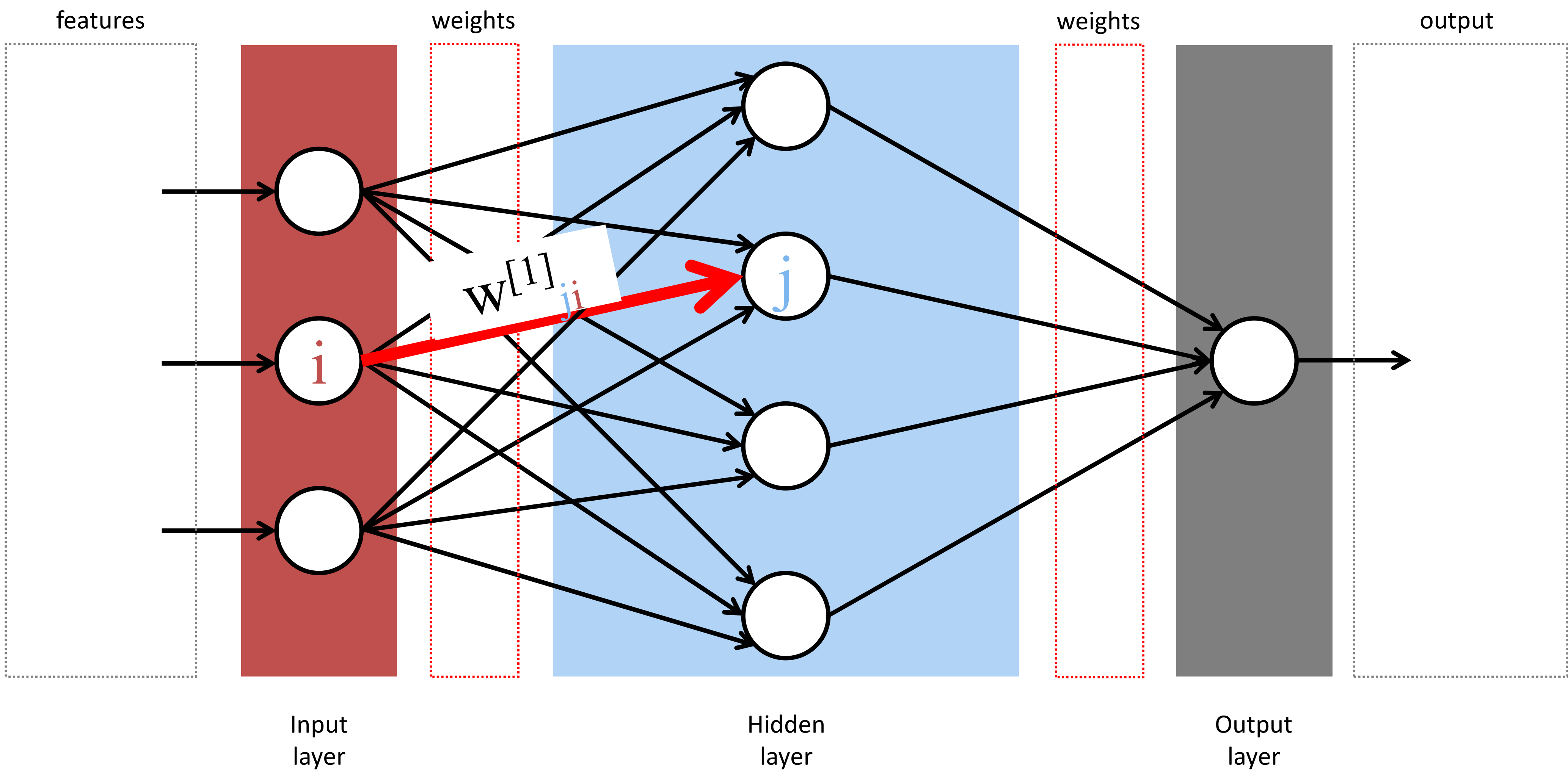
Size:

175 billion machine learning parameters

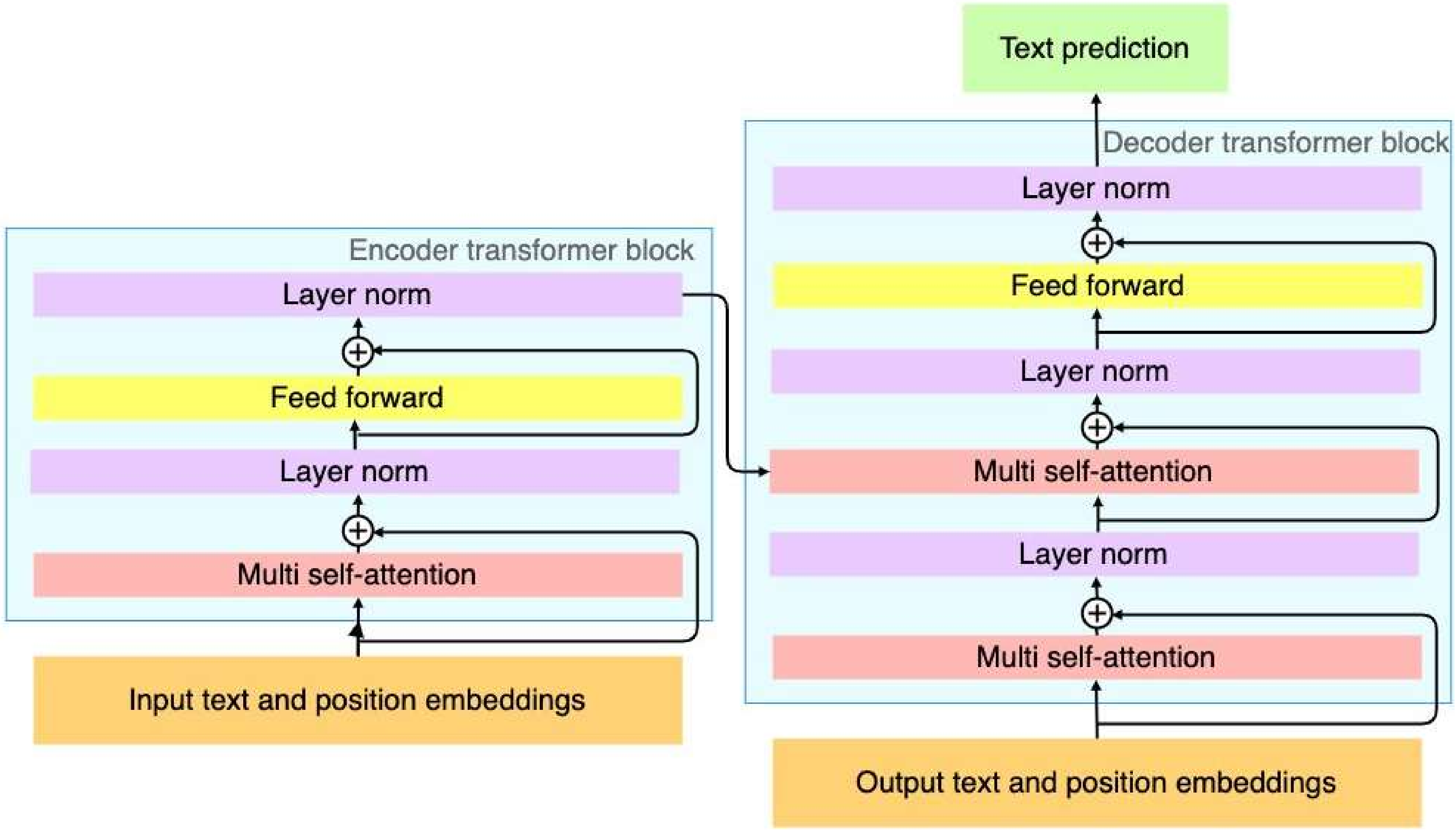
~45 GB

Source: Wikipedia

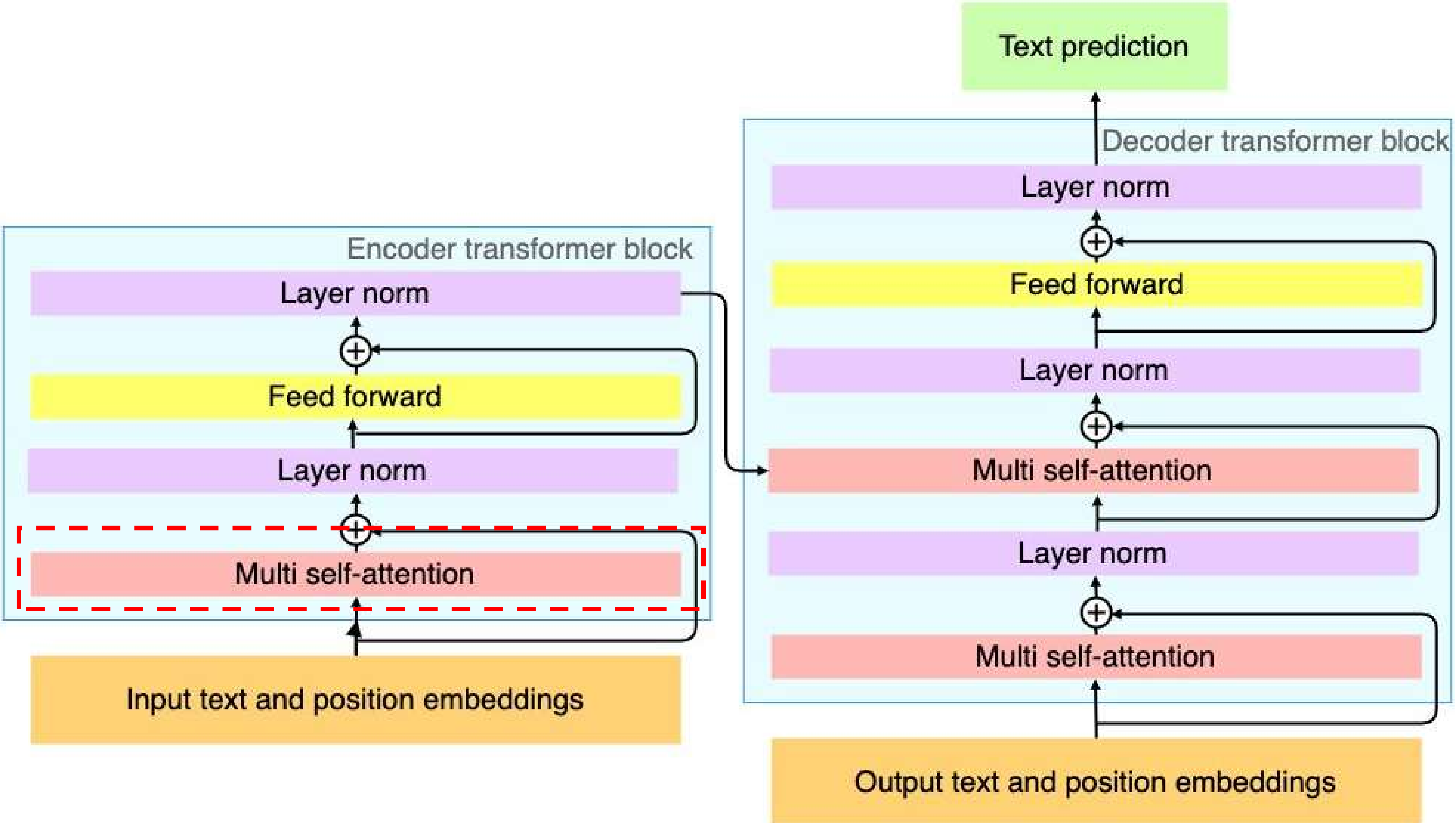
Parameters? What Are Those?



Transformer Architecture

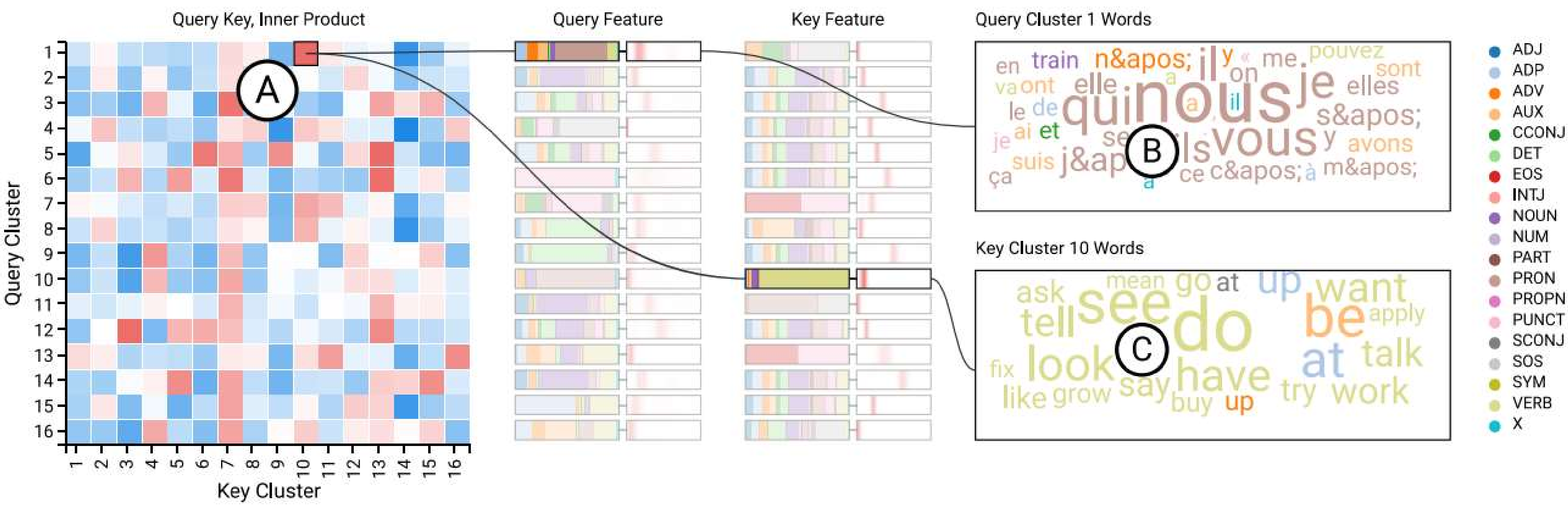


Transformer Architecture



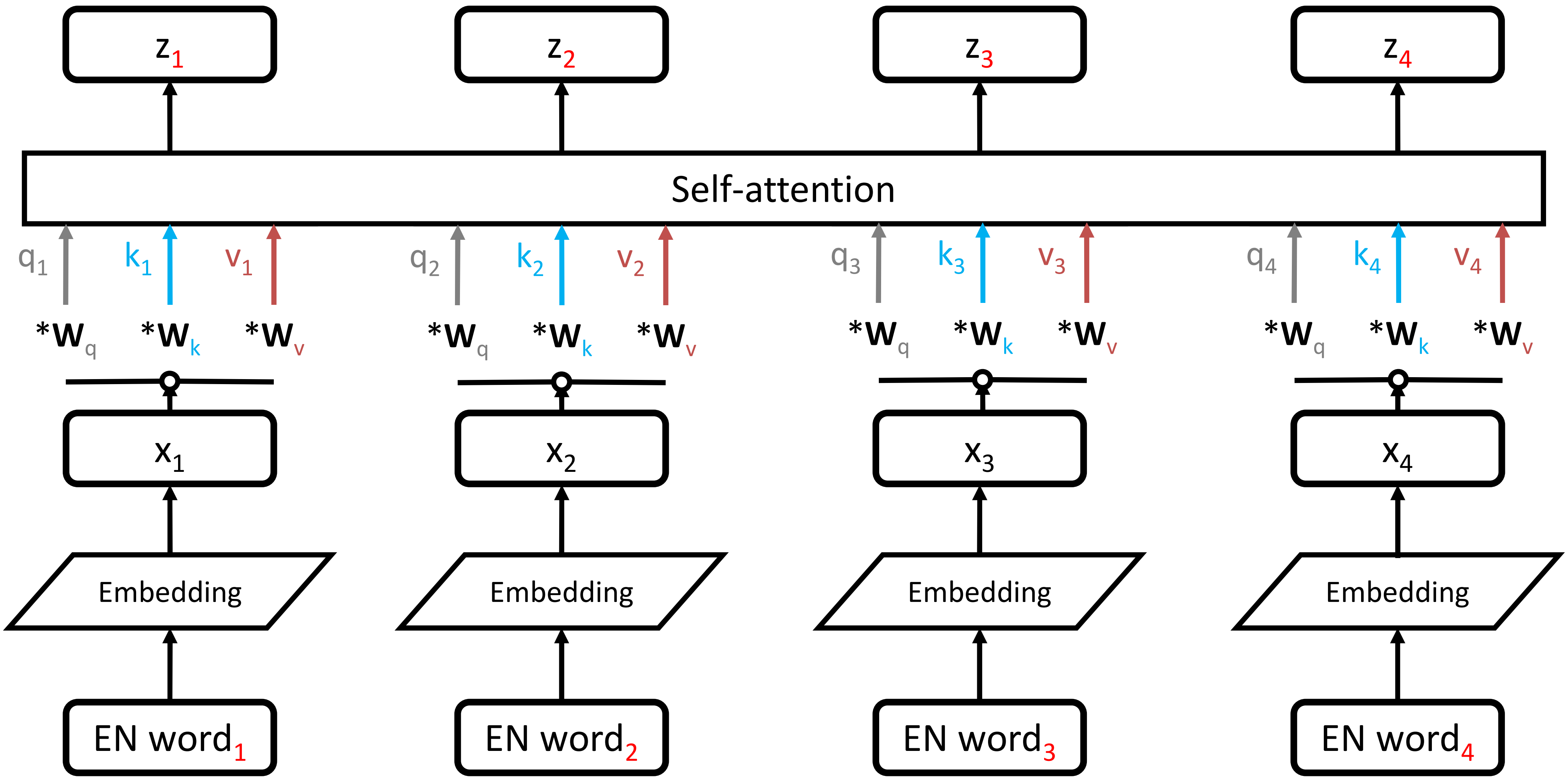
Self-Attention

In artificial neural networks, **attention is a technique that is meant to mimic cognitive attention**. The effect **enhances some parts of the input data while diminishing other parts** — the motivation being that the network should devote more focus to the important parts of the data, even though they may be small. **Learning which part of the data is more important than another depends on the context**, and this is trained by gradient descent.



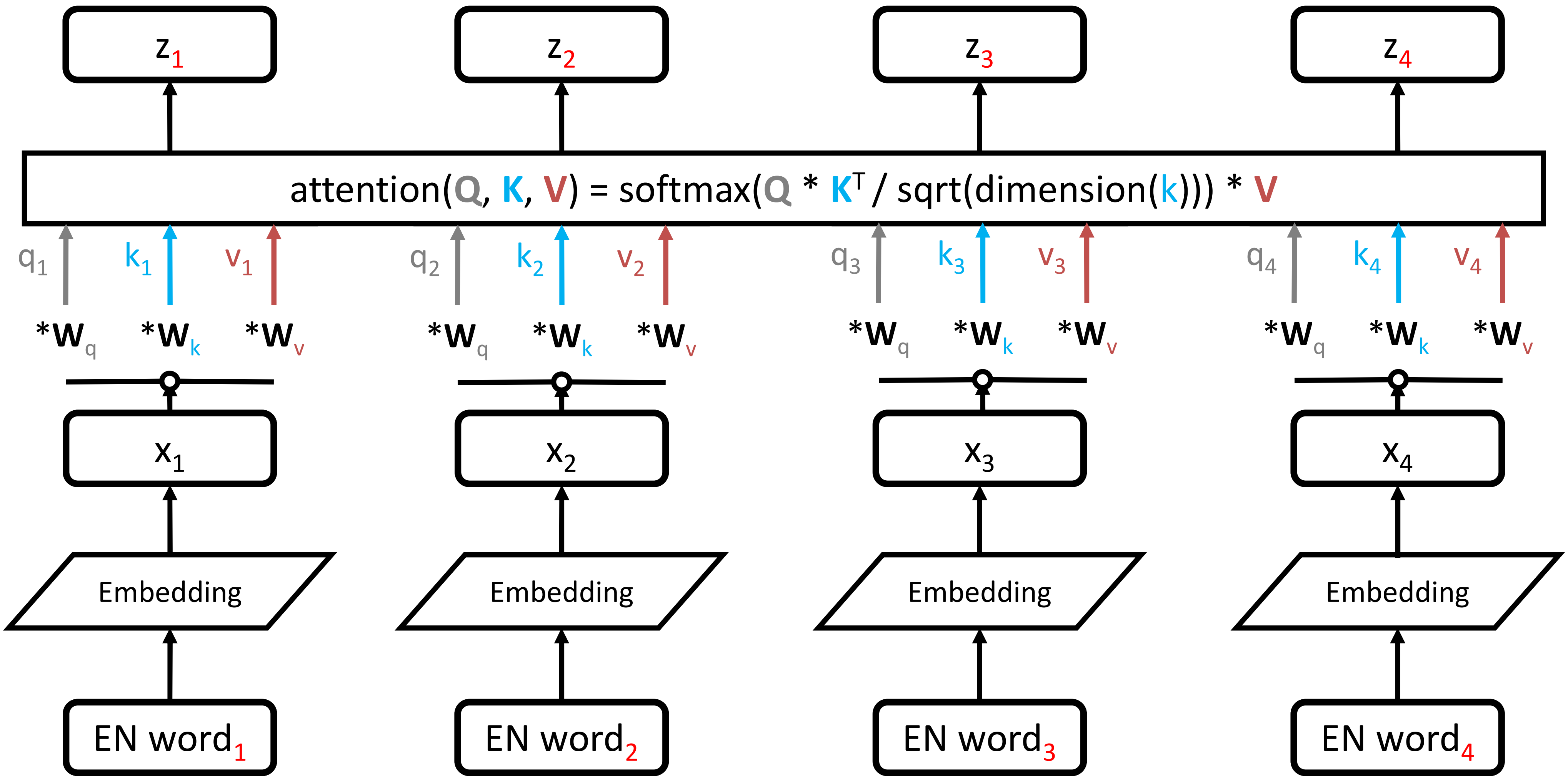
Source: Park et al. – "SANVis: Visual Analytics for Understanding Self-Attention Networks"

Single Head Self-Attention



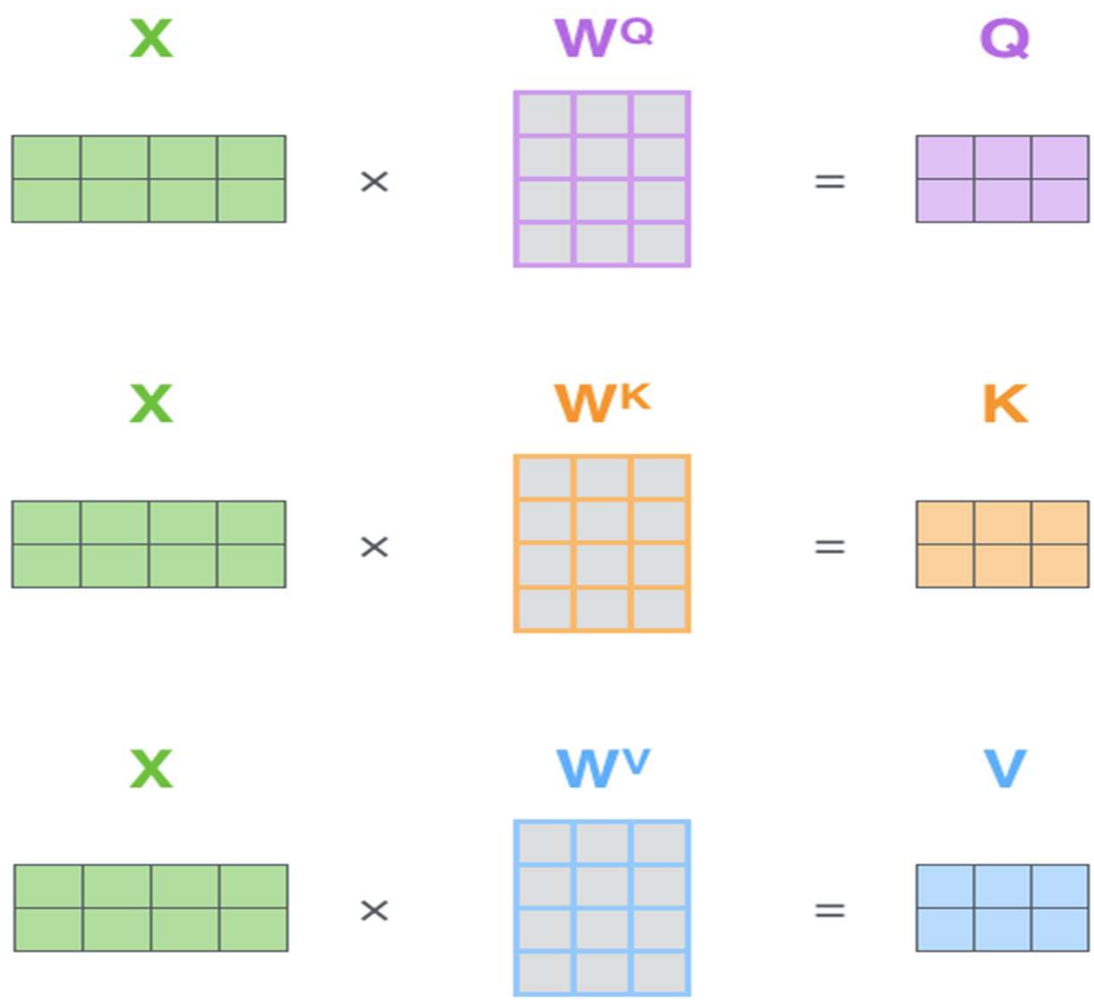
q_i, k_i, v_i – query$_i$, key$_i$, value$_i$ | W_q, W_k, W_v - query, key, value weight matrices [trained: backpropagation]

Single Head Self-Attention



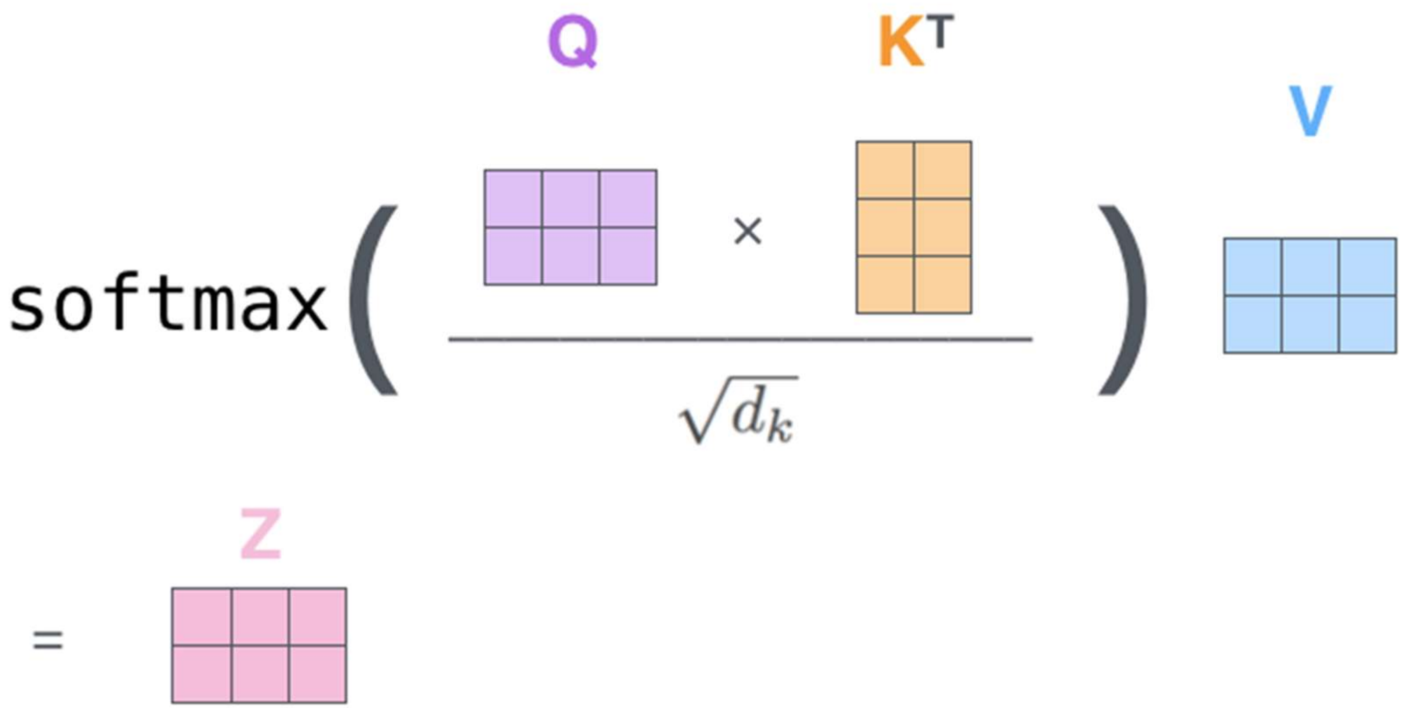
q_i, k_i, v_i – $query_i$, key_i, $value_i$ | W_q, W_k, W_v - query, key, value weight matrices [trained: backpropagation]

Single Head Self-Attention



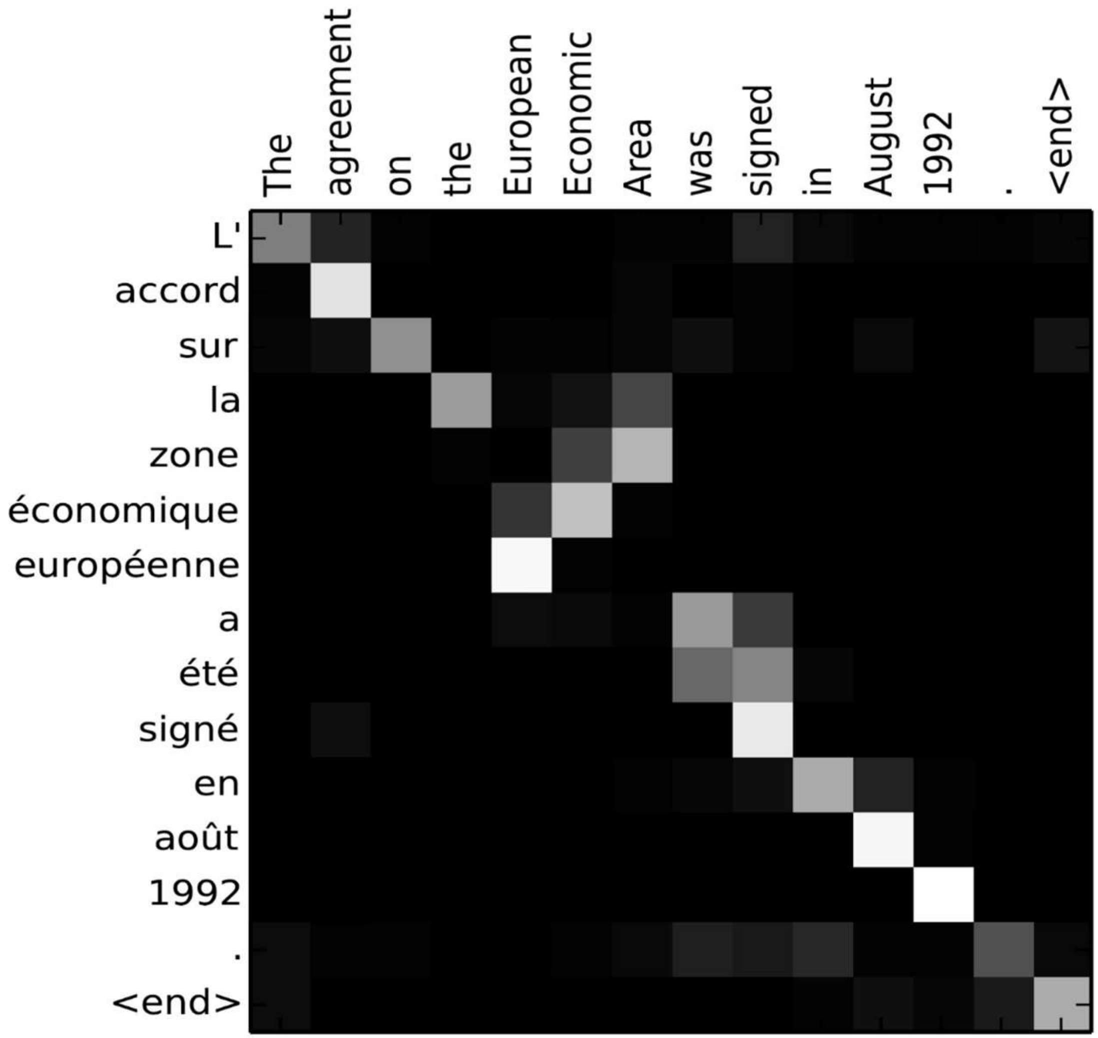
source: https://jalammar.github.io/illustrated-transformer/

Single Head Self-Attention



source: https://jalammar.github.io/illustrated-transformer/

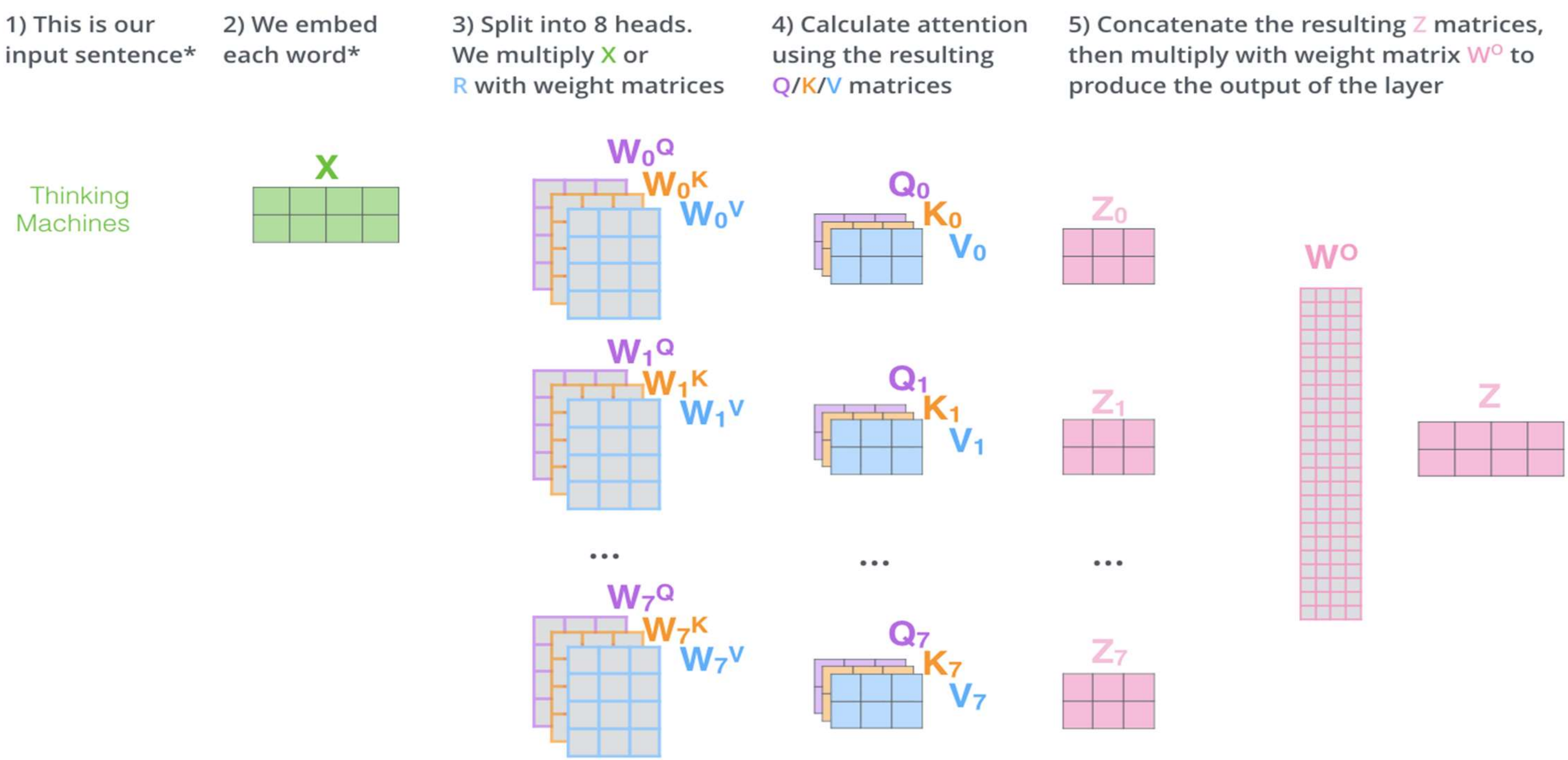
Alignment Matrix



source: https://arxiv.org/pdf/1409.0473.pdf

Multi-Head Self-Attention

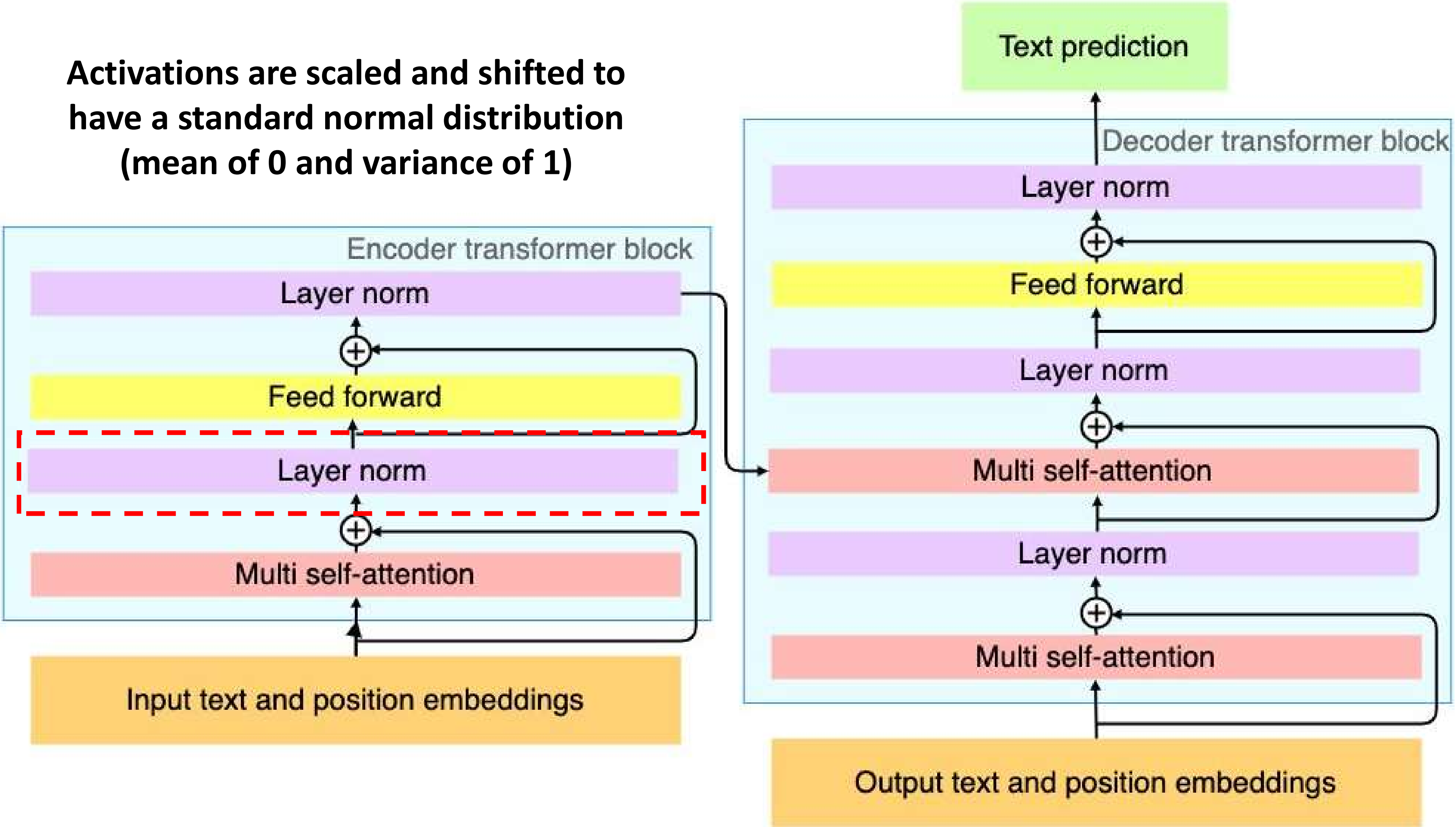
$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1}, ..., \mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



source: https://jalammar.github.io/illustrated-transformer/

Transformer Architecture

Activations are scaled and shifted to have a standard normal distribution (mean of 0 and variance of 1)



Generative Pre-trained Transformer 4

What is it?

Generative Pre-trained Transformer 4 (GPT-4) is a **multimodal large language model** created by OpenAI. As a transformer, GPT-4 was **pretrained to predict the next token** (using both public data and "data licensed from third-party providers"), and was then **fine-tuned with reinforcement learning from human and AI feedback for human alignment and policy compliance**.

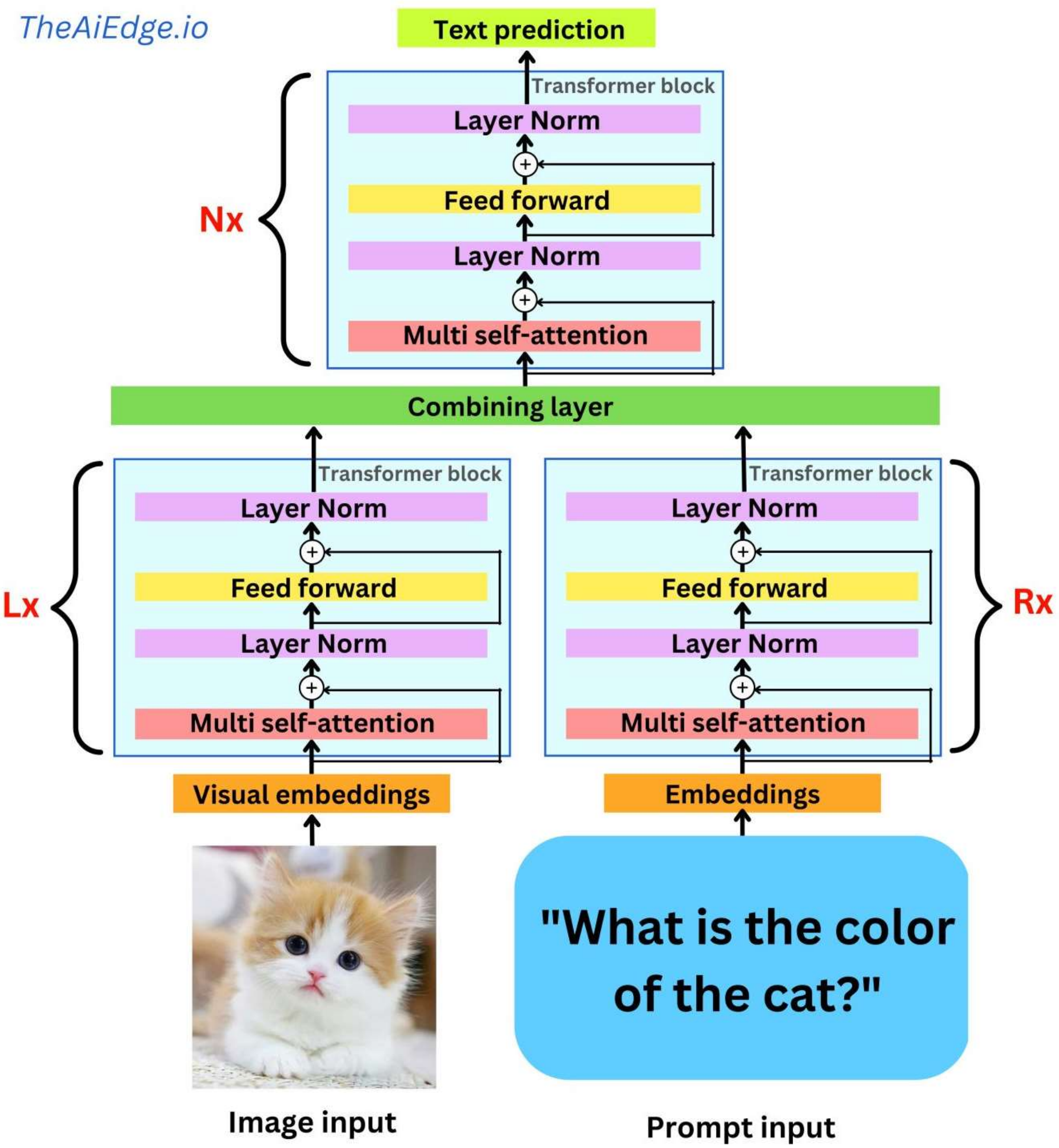
Size:

1 trillion machine learning parameters

~45 GB

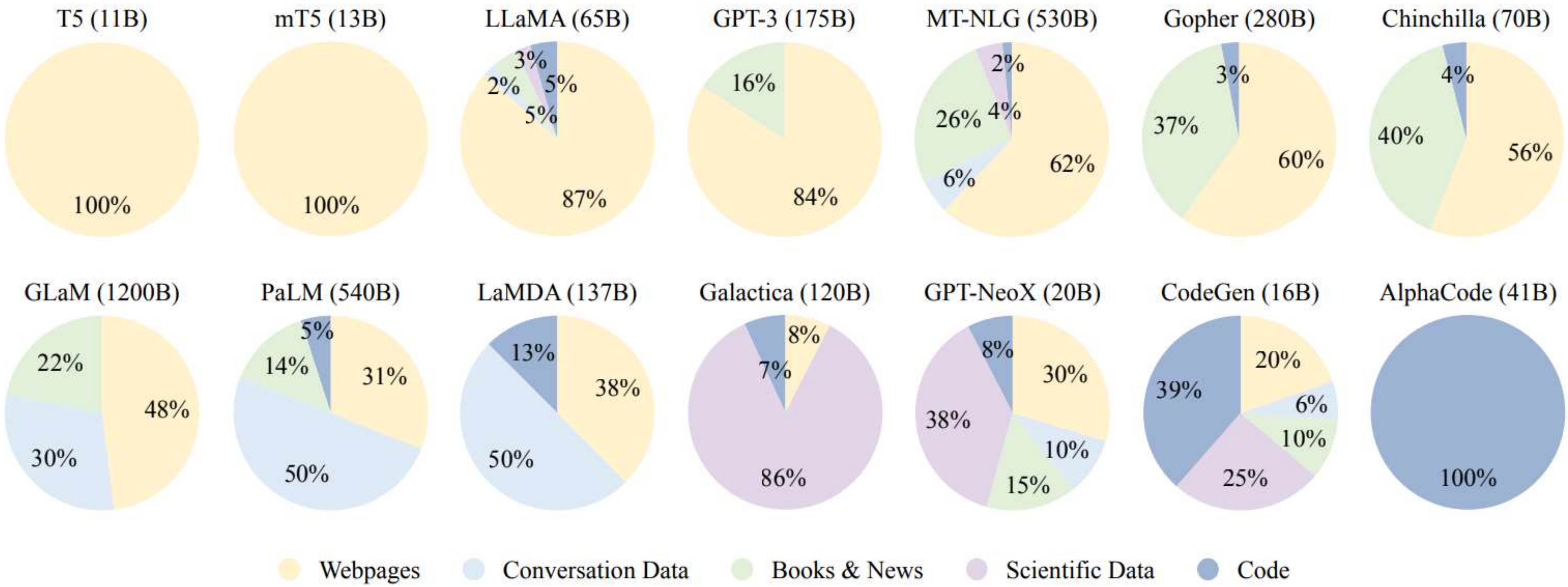
Source: Wikipedia

GPT-4 Architecture



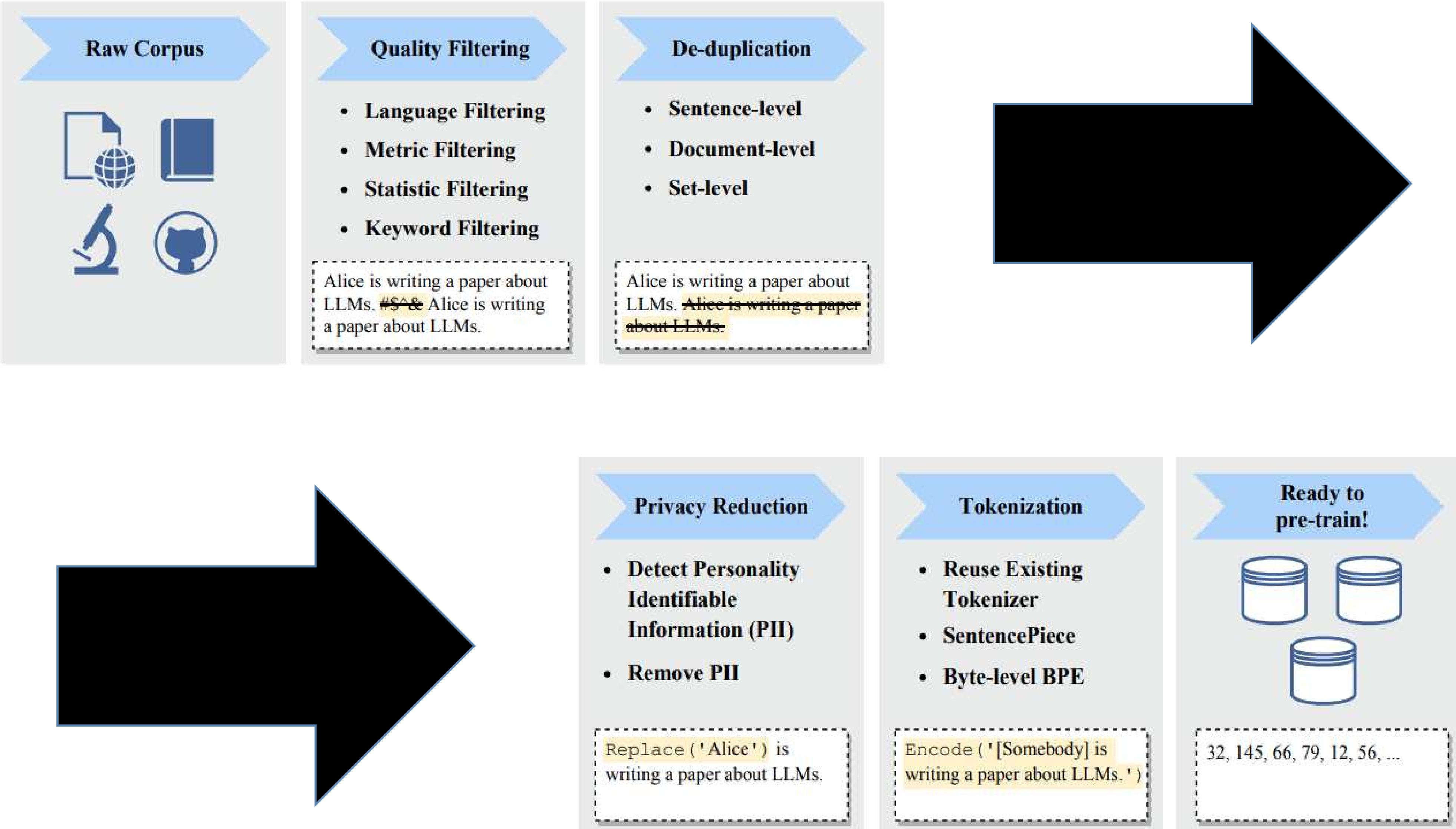
Source: TheAiEdge.io

Large Language Models Data Sources



Source: Zhao et al. – "A Survey of Large Language Models" [2023]

LLM Data Pre-Processing Pipeline



Source: Zhao et al. – "A Survey of Large Language Models" [2023]

ChatGPT

What is it?

ChatGPT is a **chatbot** developed by OpenAI and released in November 2022. It is **built on top of OpenAI's GPT-3.5 and GPT-4 families of large language models** (LLMs) and has been **fine-tuned** (an approach to **transfer learning**) **using both supervised and reinforcement learning techniques**.

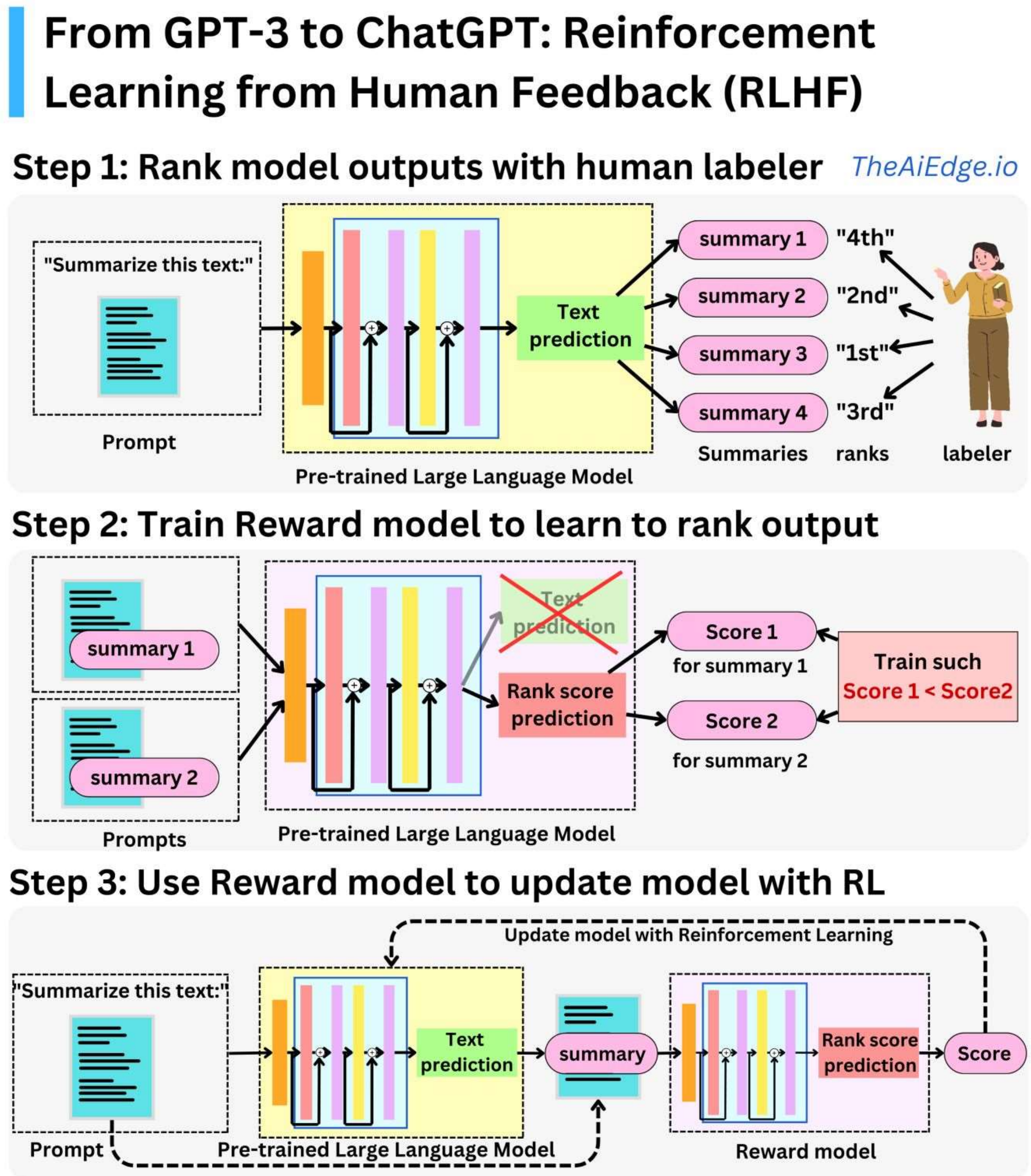
Source: Wikipedia

Transfer Learning

In transfer learning, experience with one learning task helps an agent learn better on another task.

Pre-trained models can be used as a starting point for developing new models.

ChatGPT: Learning From Feedback

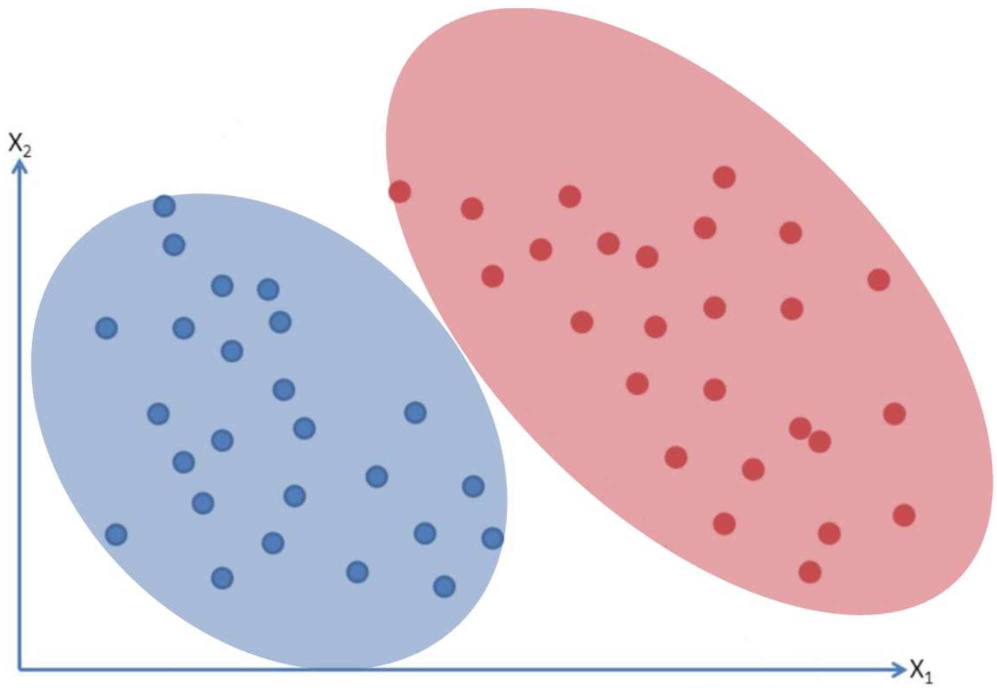


Source: TheAiEdge.io

Generative Models

Generative vs Discriminative Models

Generative

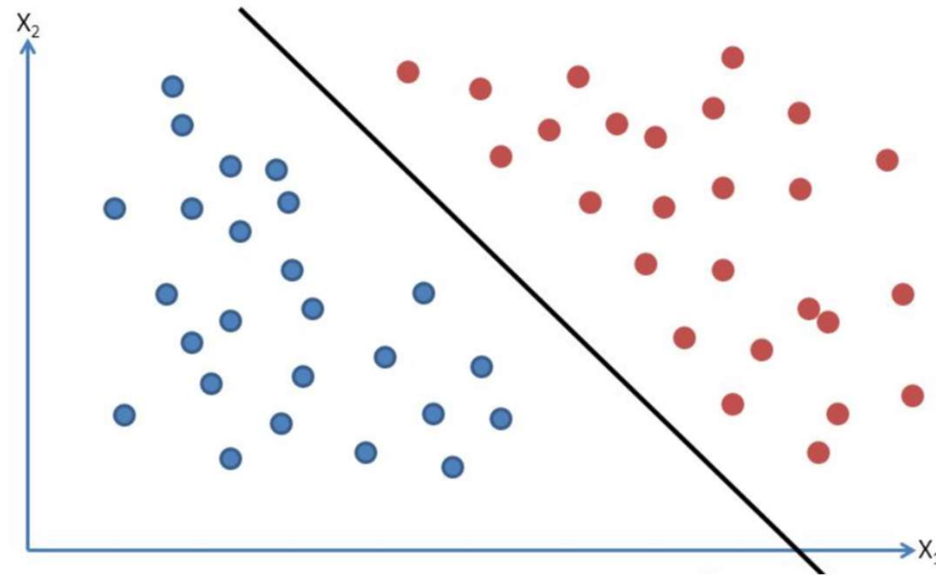


Generative model models actual distributions for EACH CLASS / LABEL / TAG

to

make a $P(\text{class} \mid \text{sample})$ **prediction**

Discriminative



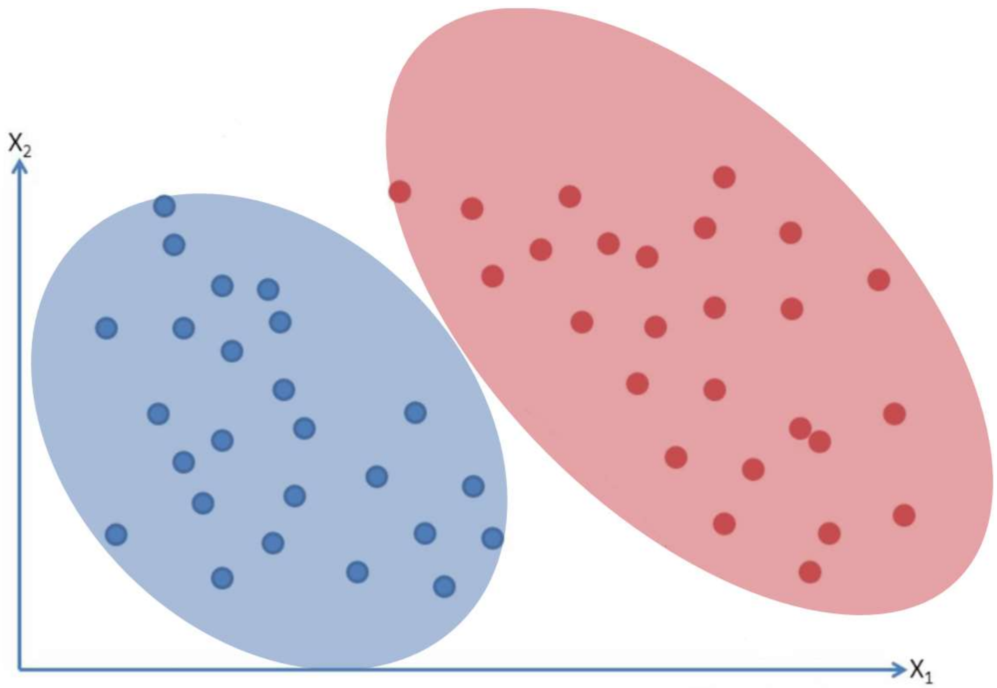
Discriminative model models the decision boundary between CLASSES / LABELS / TAGS

to

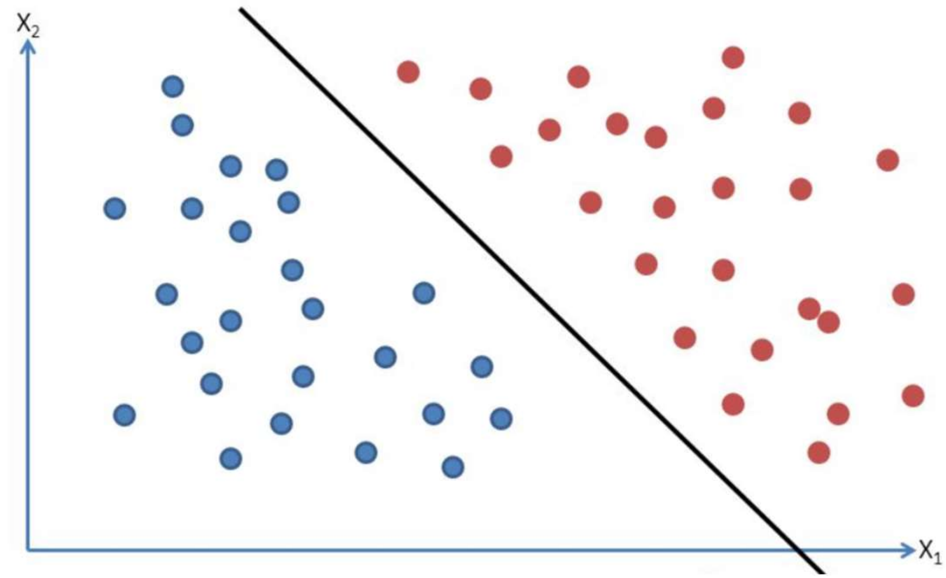
make a $P(\text{class} \mid \text{sample})$ **prediction**

Generative vs Discriminative Models

Generative



Discriminative



Generative model uses training data to learn P(sample, class) **joint probabilities**

and then

uses Bayes Theorem to get the P(class | sample) **prediction**

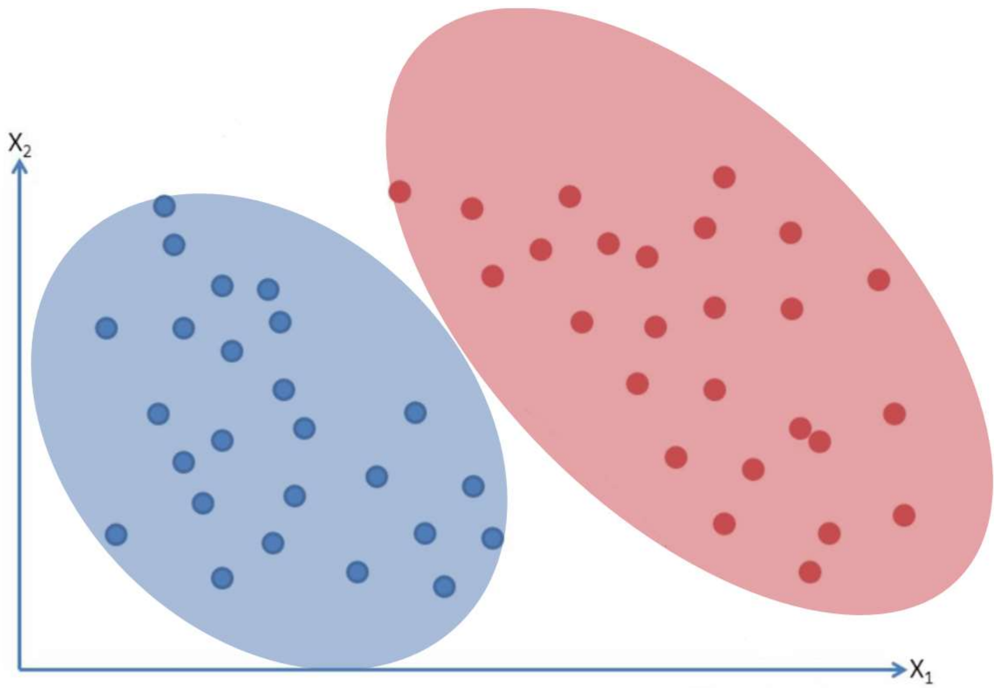
Discriminative model uses training data to learn P(class | sample) **conditional probability**

and then

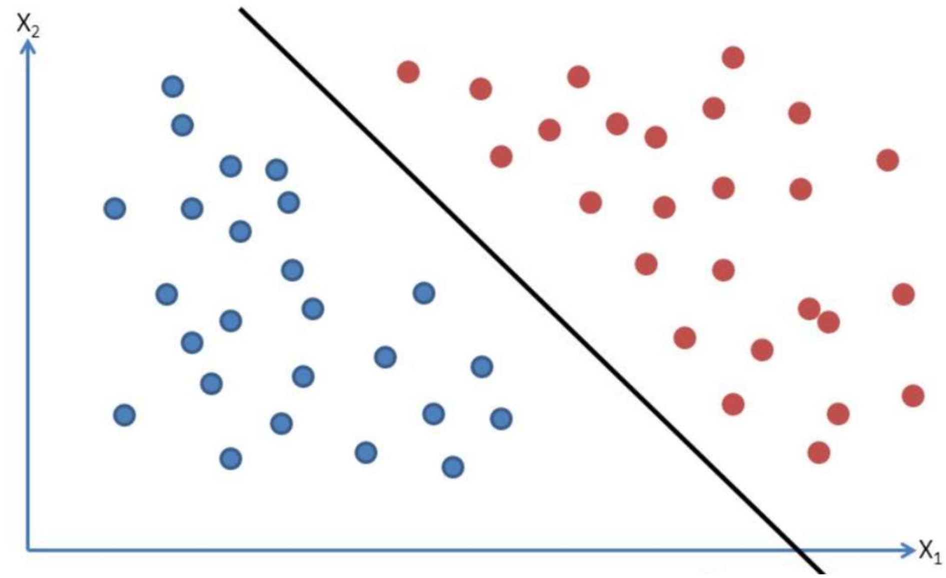
uses it to make a prediction

Generative vs Discriminative Classifier

Generative



Discriminative



Generative classifiers:

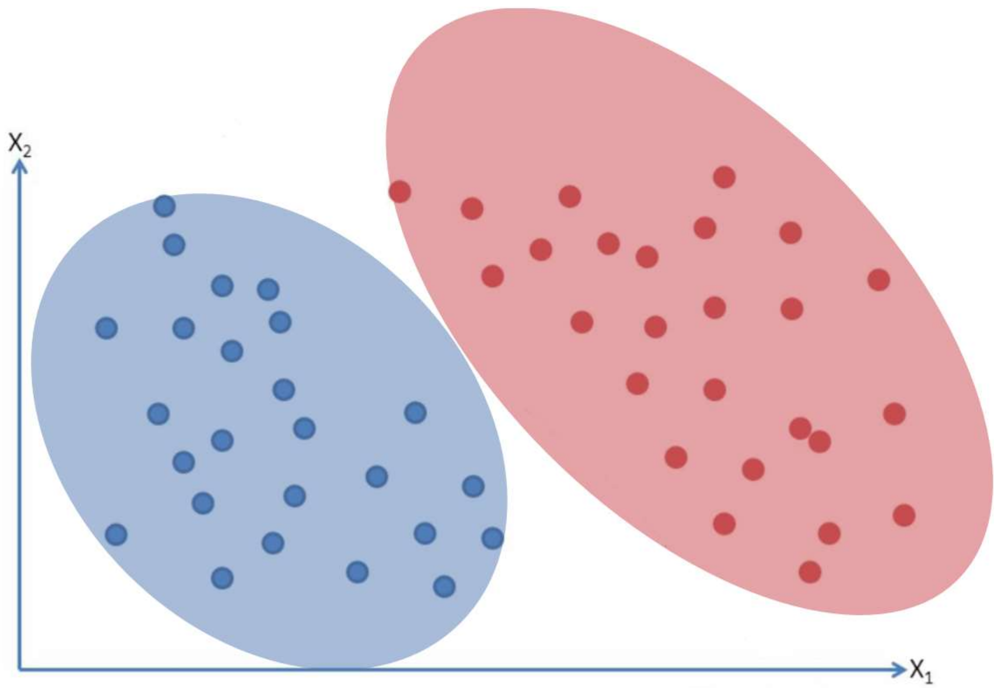
- **Assume some form of** P(class)**,** P(sample | class)
- **Estimate** P(class)**,** P(sample | class) **using training data**
- **Use Bayes Theorem to calculate** P(class | sample)

Discriminative classifiers:

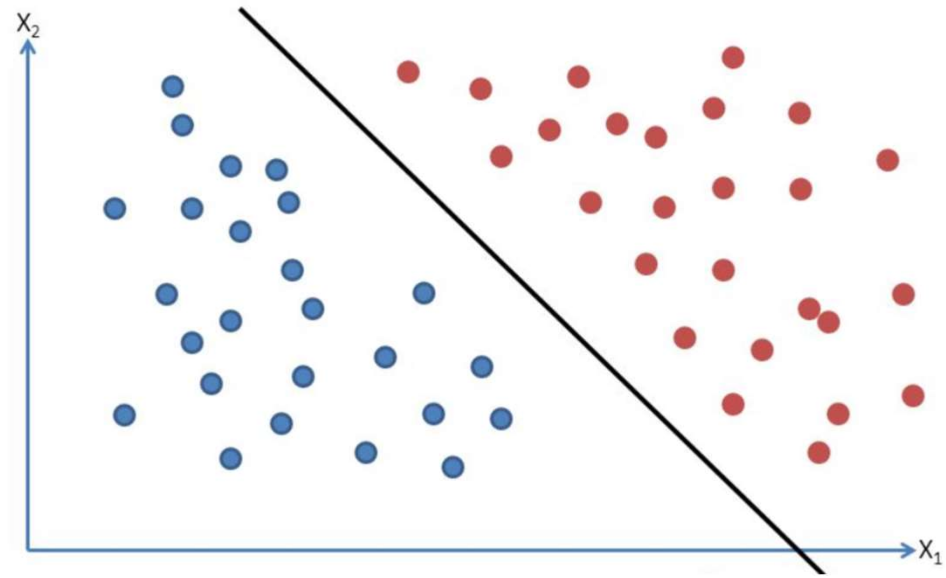
- **Assume some form of** P(class | sample)
- **Estimate** P(class | sample) **using training data**

Generative vs Discriminative Classifier

Generative



Discriminative



Generative classifiers:
- Naive Bayes
- Bayesian networks
- Markov random fields
- Hidden Markov Models (HMM)

Discriminative classifiers:
- Logistic regression
- Support Vector Machines
- Traditional neural networks
- k-Nearest Neighbors
- Conditional Random Fields (CRF)s

Generative AI Model: the Idea

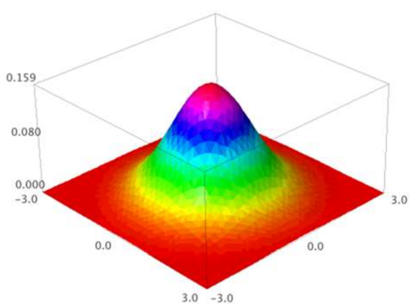
Reality: some sample space (possibly infinite)

Sample

Dataset: Subset of “Reality” (finite)

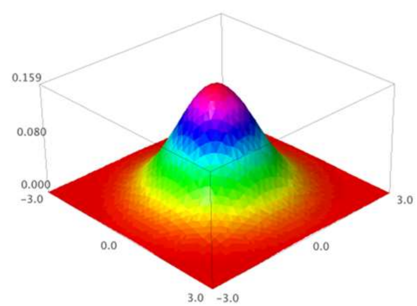
Model

New sample



Probability distribution P_{real}

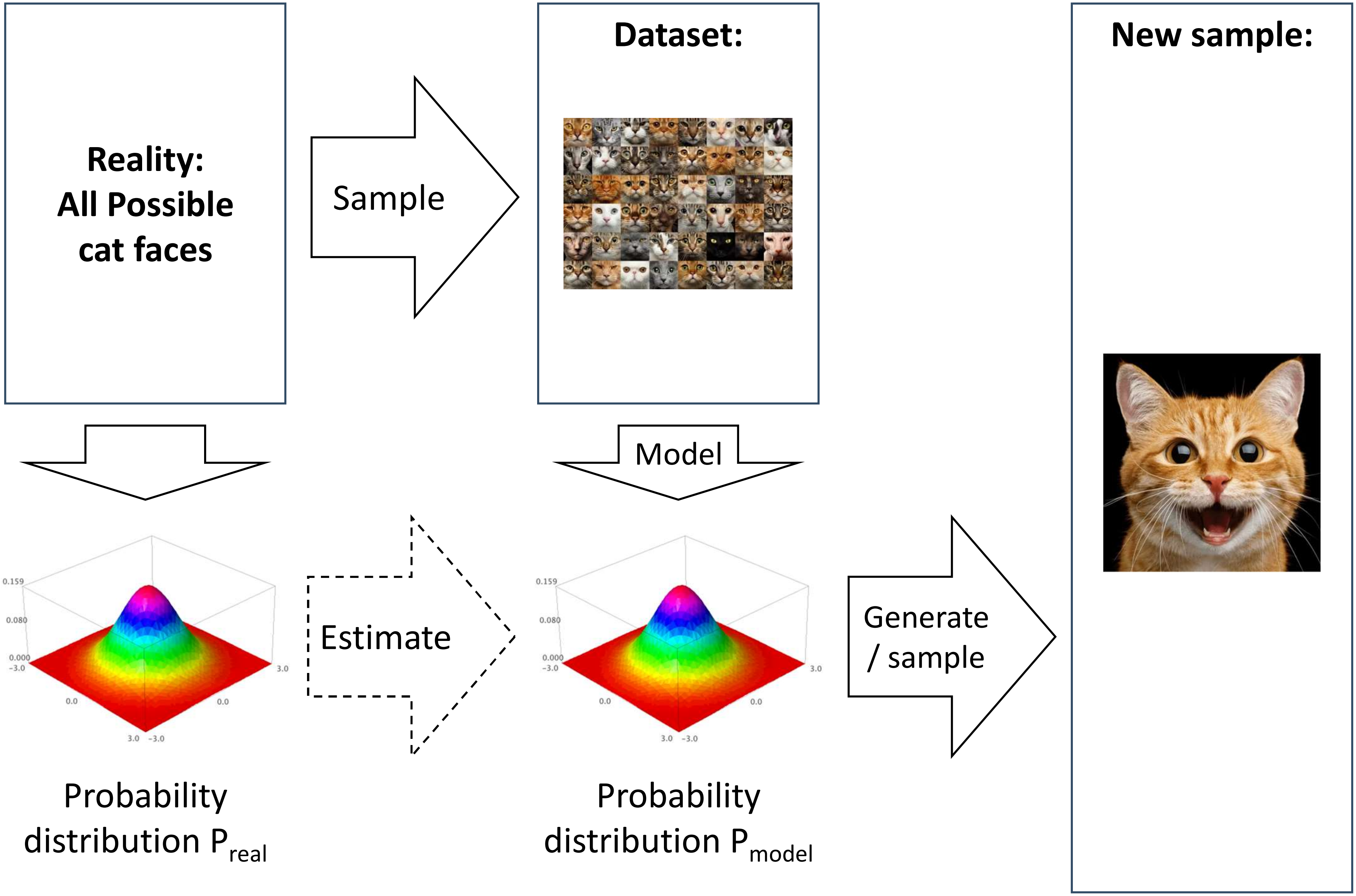
Estimate



Probability distribution P_{model}

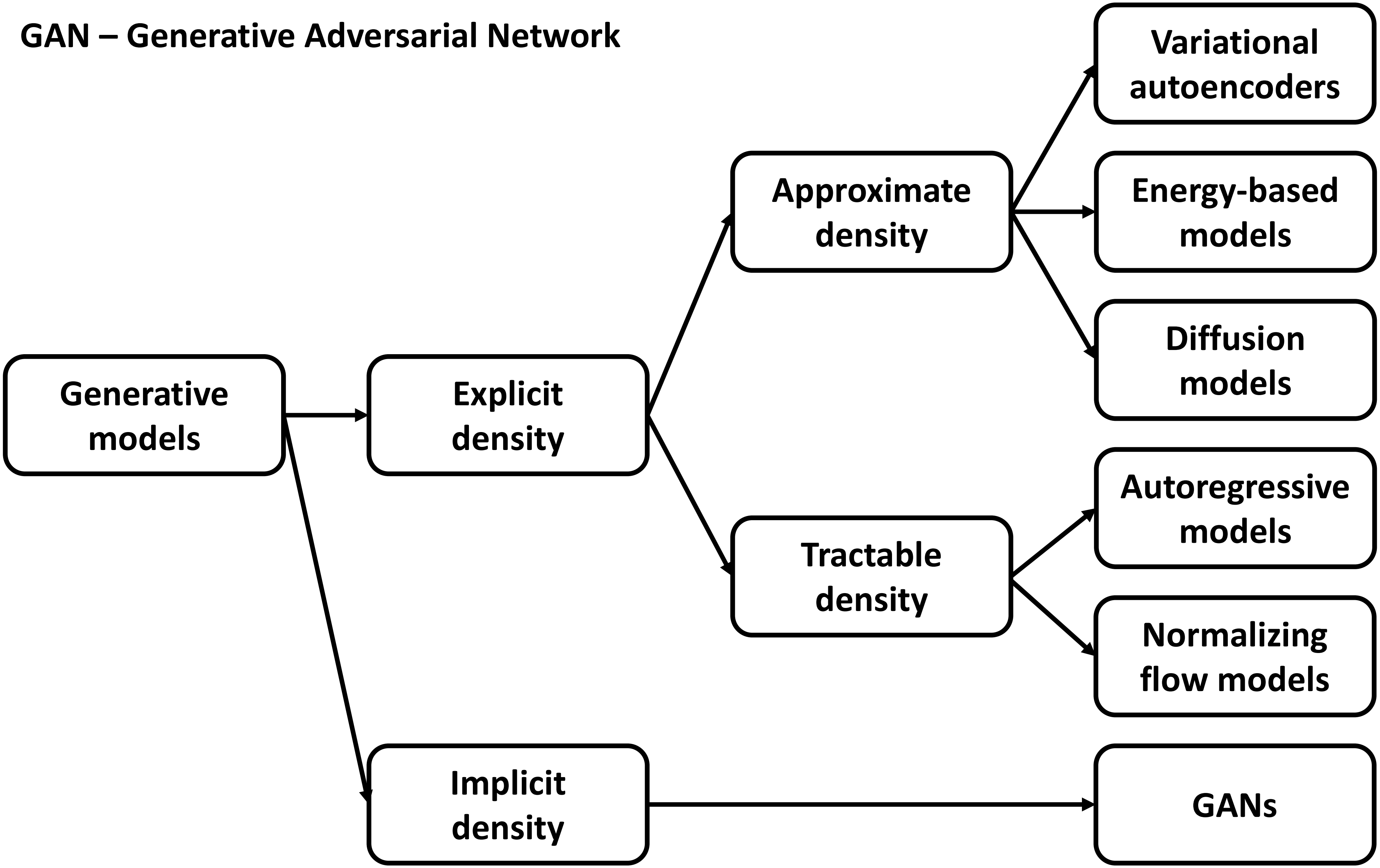
Generate / sample

Generative AI Model: the Idea



Taxonomy of Generative AI Models

GAN – Generative Adversarial Network



Explicit Density

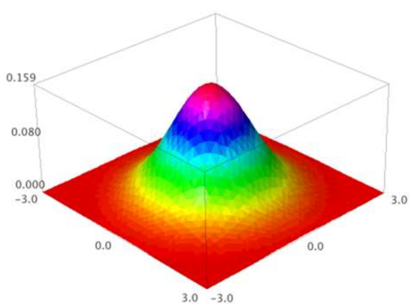
Reality:
All Possible
cat faces

Sample

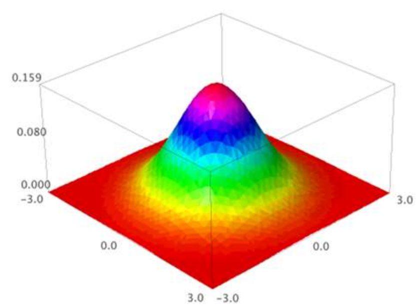
Dataset:

New sample:

Model



Estimate



Sample

Probability distribution P_{real}

Probability distribution P_{model}

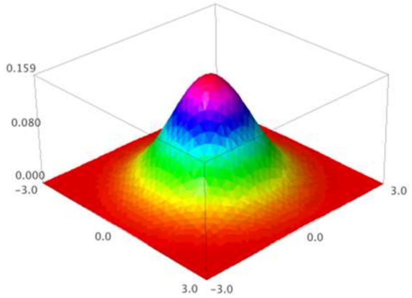
Implicit Density

Reality:
All Possible cat faces

Sample

Dataset:

Model



Probability distribution P_{real}

Stochastic generation process

Generate

New sample:

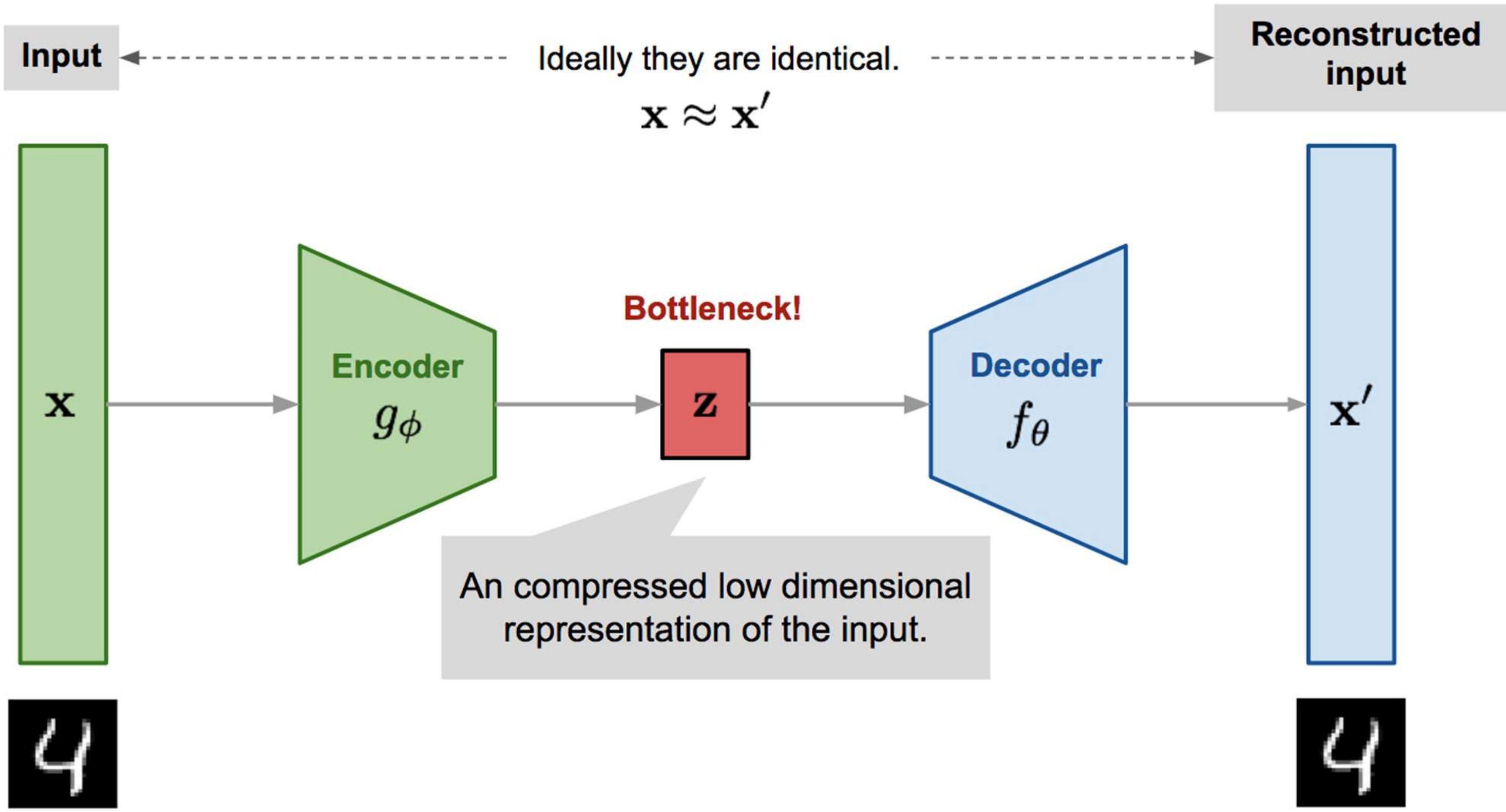
Tractable vs. Approximate Density

Tractable density models place constraints on the model architecture so that the density function has a form that makes it easy to calculate.

Approximate density models use variety of techniques to approximate the density function:

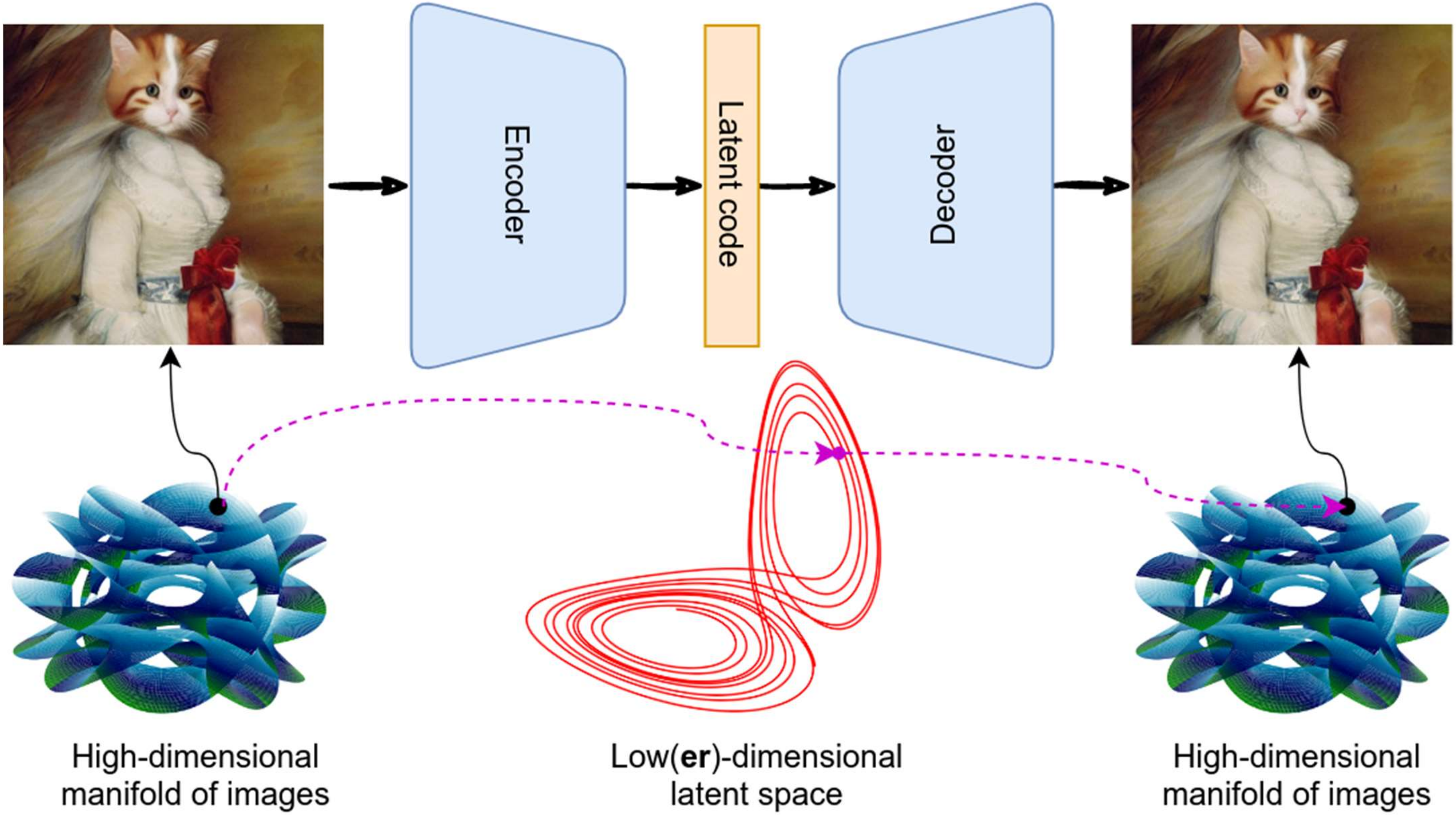
- **latent vectors**
- **denoising**

Autoencoder Model



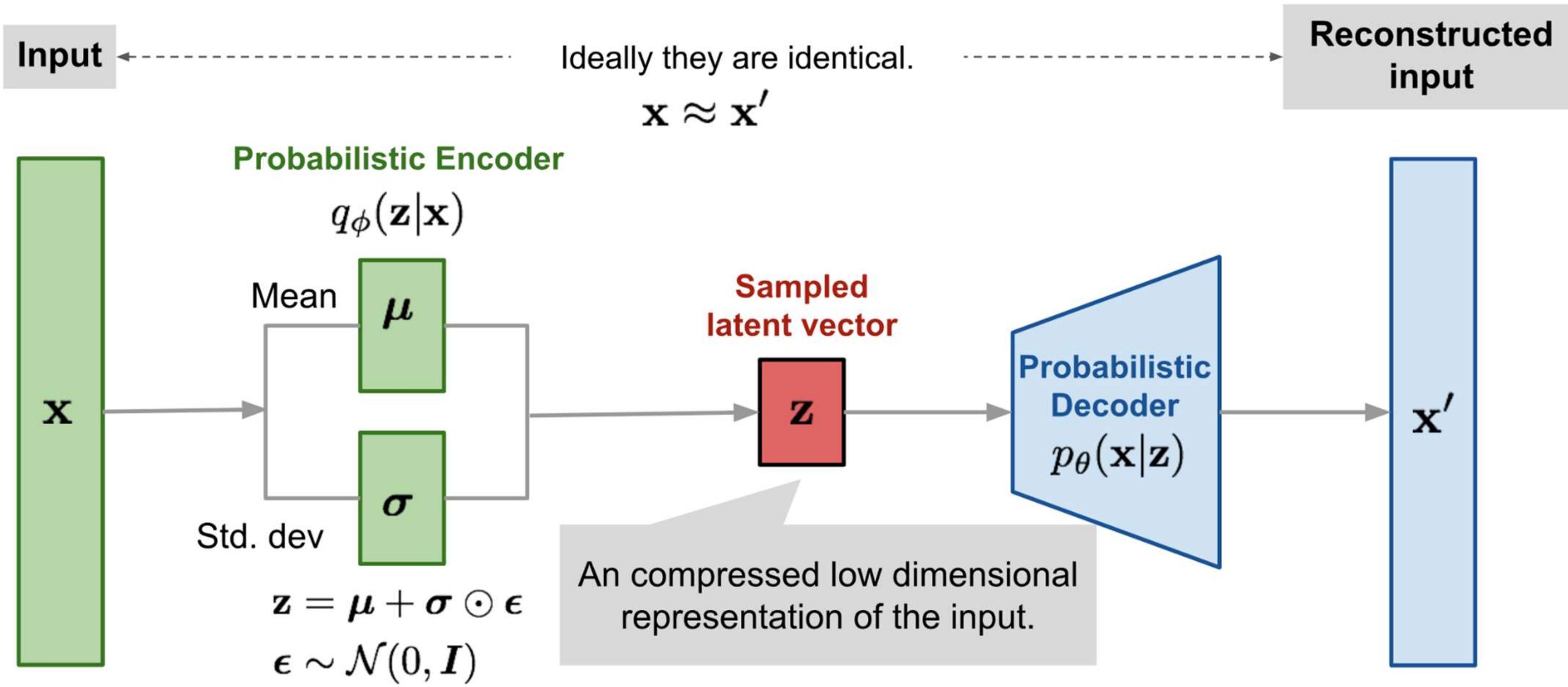
Source: https://lilianweng.github.io/posts/2018-08-12-vae/

Latent Space



Source: https://synthesis.ai/2023/03/21/generative-ai-ii-discrete-latent-spaces/

Variational Autoencoder Model



Source: https://lilianweng.github.io/posts/2018-08-12-vae/

Autoregressive Model (GPT-3)

What is it?

Generative Pre-trained Transformer 3 (GPT-3) is an **autoregressive language model that uses deep learning to produce human-like text**. It is the third-generation language prediction model in the GPT-n series (and the successor to GPT-2) created by OpenAI, a San Francisco-based artificial intelligence research laboratory.

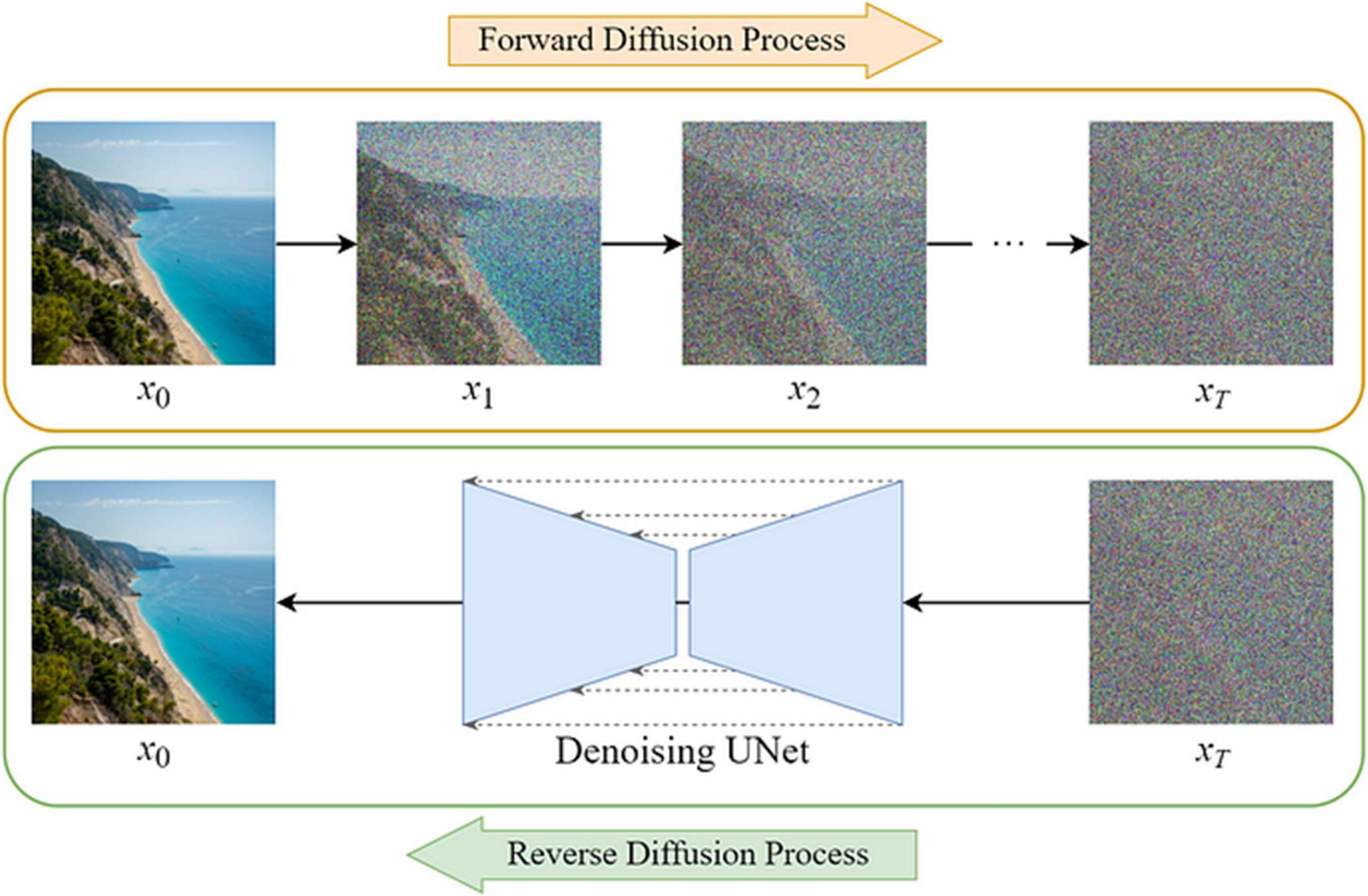
Size:

175 billion machine learning parameters

~45 GB

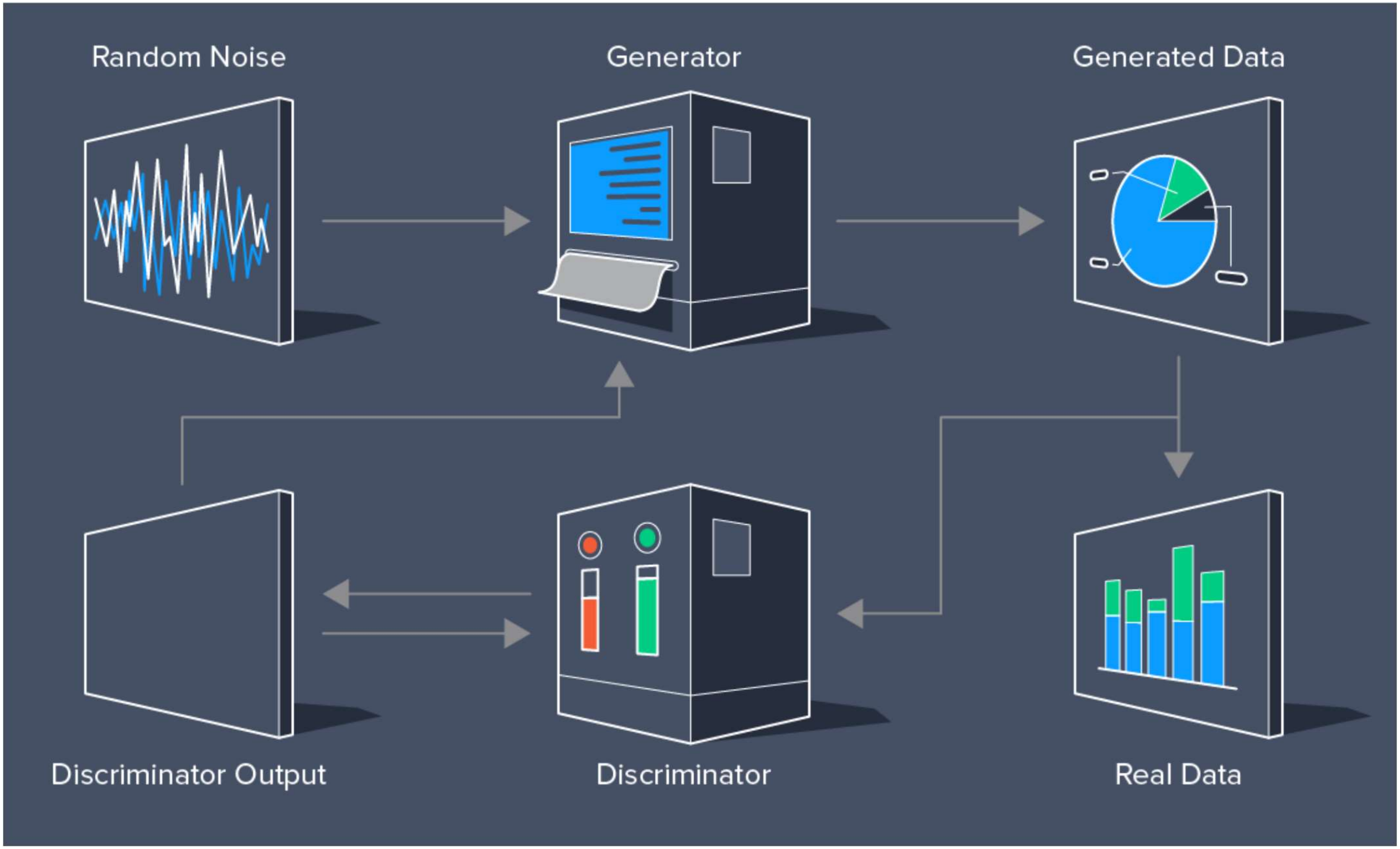
Source: Wikipedia

Diffusion Model



Source: https://medium.com/@steinsfu/stable-diffusion-clearly-explained-ed008044e07e

Generative Adversarial Network



Source: https://www.toptal.com/machine-learning/generative-adversarial-networks