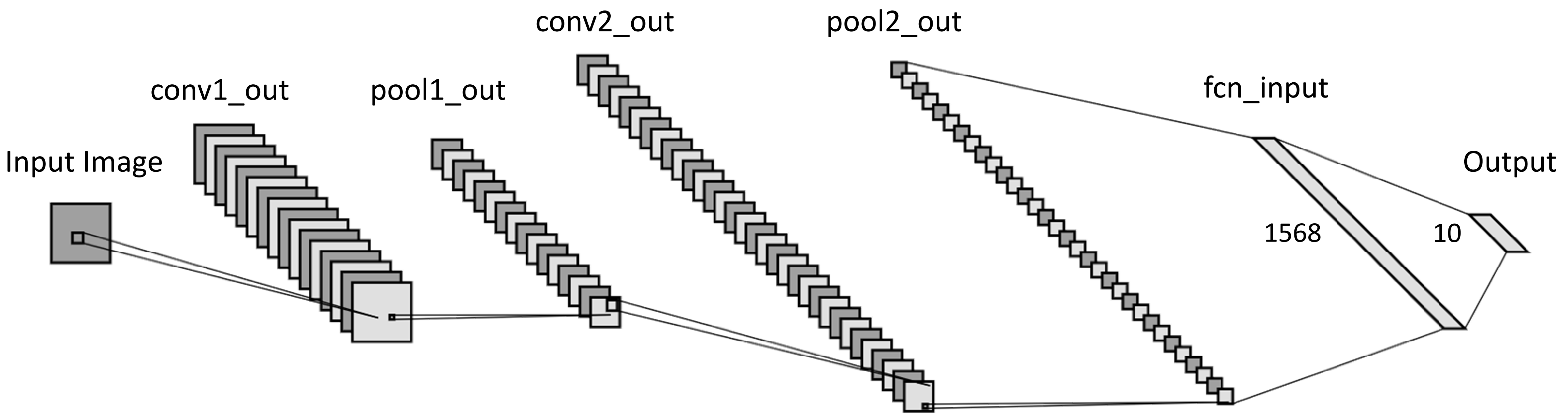
LECTURE 4:

RECURRENT NEURAL NETWORK

University of Washington, Seattle

Fall 2024

Previously in EEP 596…



self.cnn1	self.maxpool1	self.cnn2	self.maxpool2	Flatten	self.fc1
• In-channel # : 1 • Out-channel # : 16 • Kernel size : 5 • Stride = 1 • Padding = 2 • ReLU	Kernel size : 2	• In-channel # : 16 • Out-channel # : 32 • Kernel size : 5 • Stride = 1 • Padding = 2 • ReLU	Kernel size : 2		1568 → 10

OUTLINE

Part 1: Introduction to RNNs

- Why do we need RNNs?
- RNN Architecture
- Embedding and Decoder

Part 2: Training RNNs

- Backpropagation in RNNs
- Vanishing/Exploding Gradient Problem
- Training with Teacher Forcing

Part 3: RNN Problem Types

- RNN Configurations
- RNN Extensions

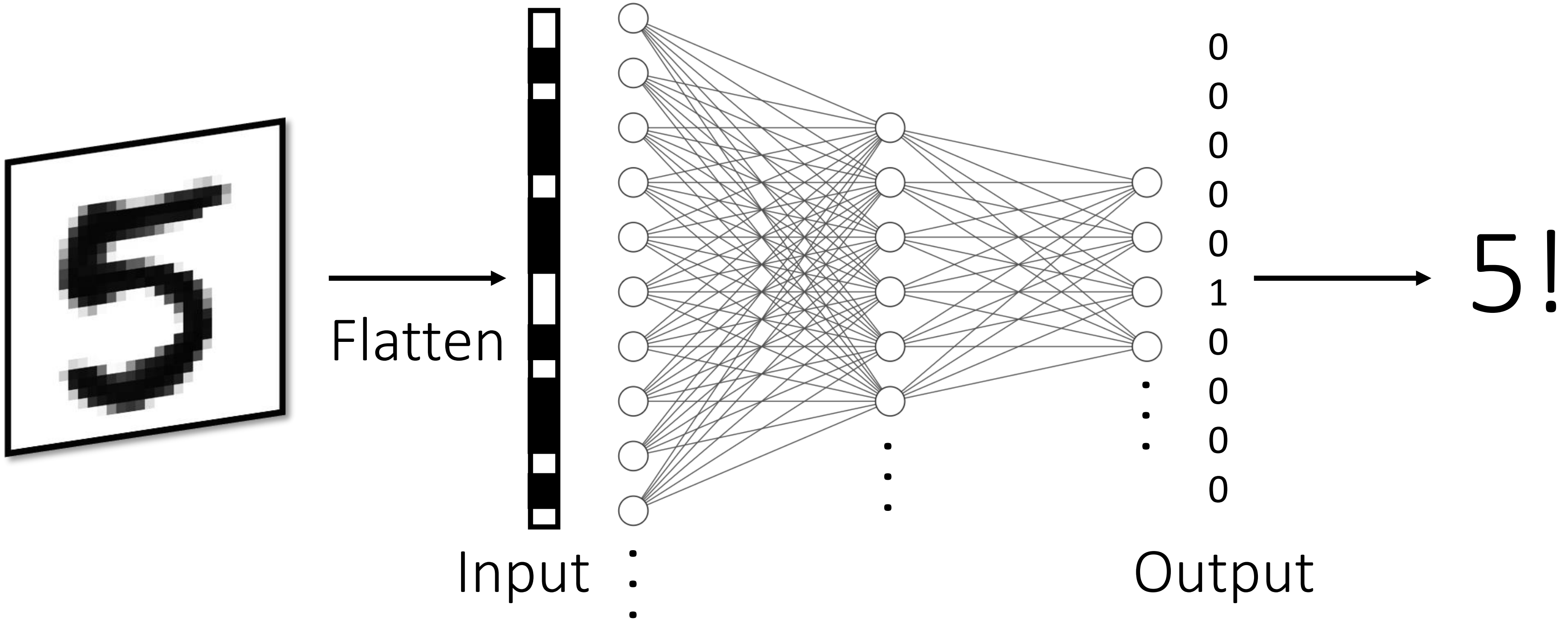
INTRODUCTION TO RNNs

Why do we need RNNs?

RNN Architecture

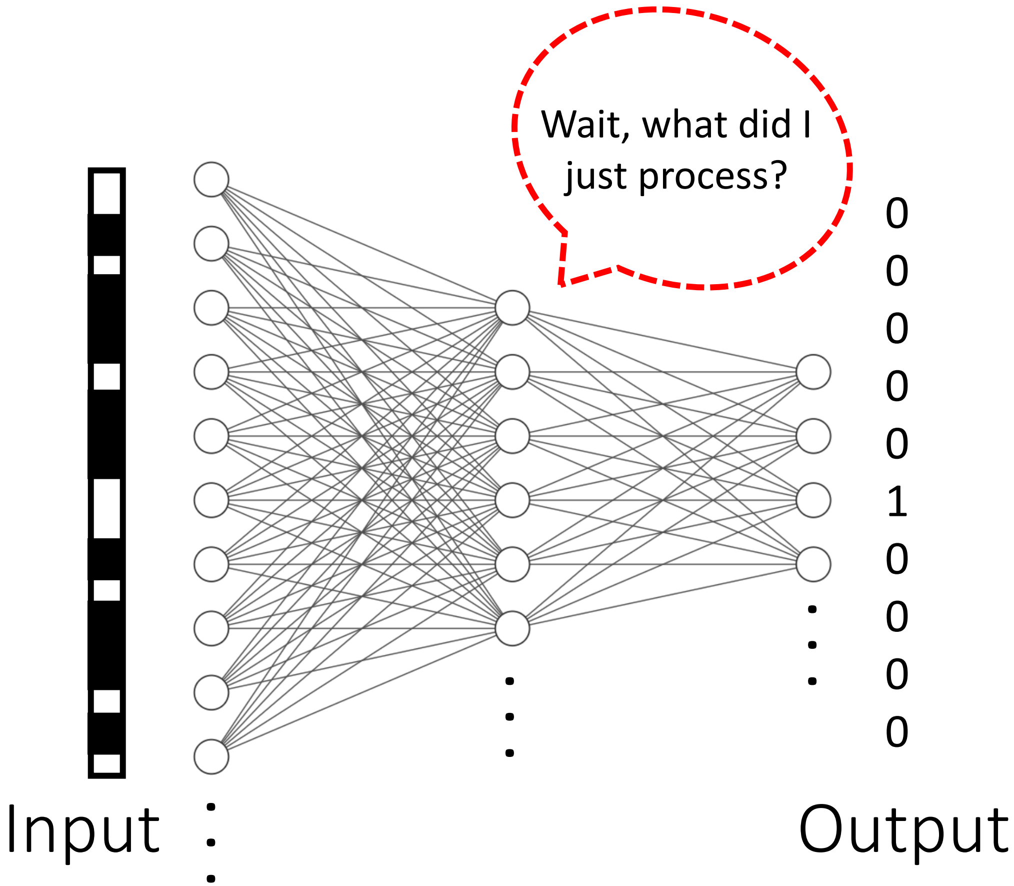
Embedding and Decoder

Why Do We Need RNNs?



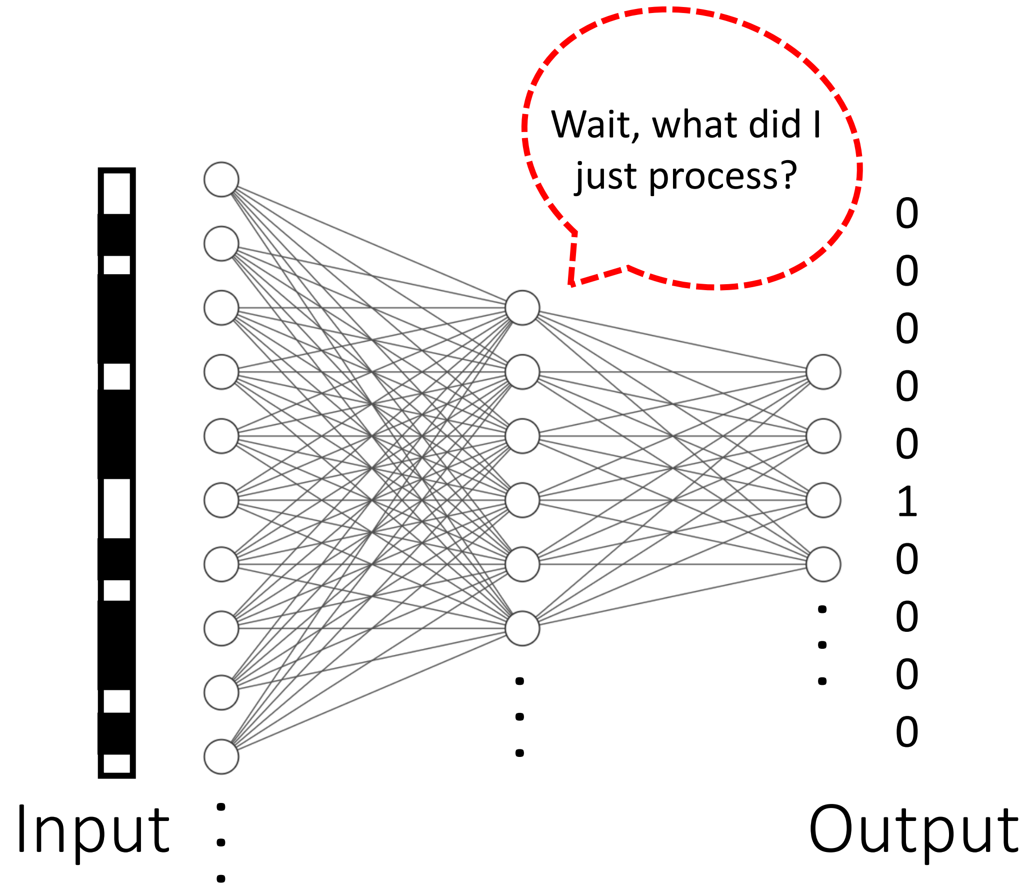
Feed-Forward Network

Why Do We Need RNNs?



Feed-Forward Network

Why Do We Need RNNs?



Feed-Forward Network has no memory of past inputs

Why Do We Need RNNs?

Korean 안녕하세요, 제 이름은 지민이에요

English Hello, my name is Jimin

Why Do We Need RNNs?

Korean	안녕하세요,
English	Hello,

Why Do We Need RNNs?

Korean 안녕하세요, 제

English Hello, my

Why Do We Need RNNs?

Korean 안녕하세요, 제 이름

English Hello, my name

Why Do We Need RNNs?

Korean 안녕하세요, 제 이름은

English Hello, my name is

Why Do We Need RNNs?

Korean 안녕하세요, 제 이름은 지민이에요

English Hello, my name is Jimin

Why Do We Need RNNs?

Korean 안녕하세요, 제 이름은 지민이에요

English Hello, my name is Jimin

Each word in a sentence is dependent to the past words → Need memory

Why Do We Need RNNs?

Korean 안녕하세요, 제 이름은 지민이에요, 그리고 저는 비디오게임을 좋아해요

English Hello, my name is Jimin, and I like videogames

A sentence (input) could have **different sizes**

Why Do We Need RNNs?

We need a neural network architecture that can handle:

Why Do We Need RNNs?

We need a neural network architecture that can handle:

- Data order

Why Do We Need RNNs?

We need a neural network architecture that can handle:

- Data order
- Temporal dependencies

Why Do We Need RNNs?

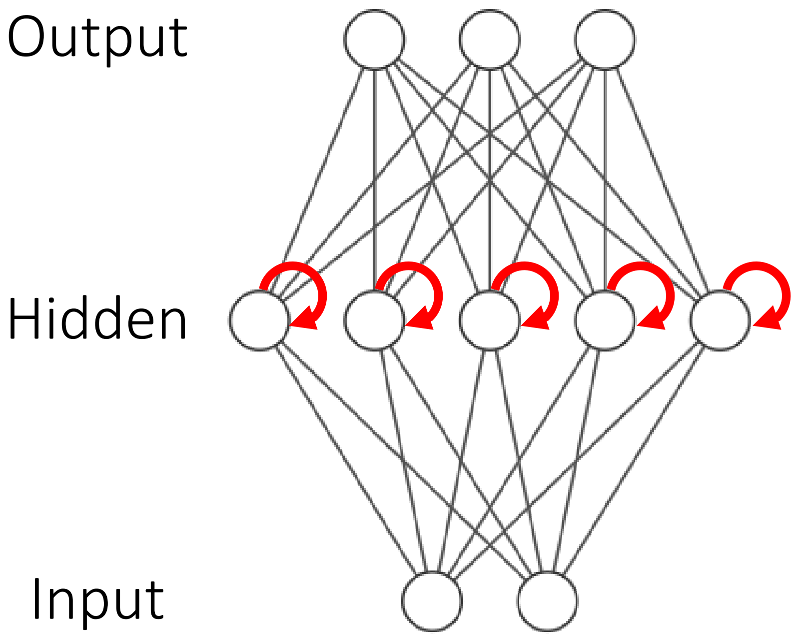
We need a neural network architecture that can handle:

- Data order
- Temporal dependencies
- Variable input sizes

Why Do We Need RNNs?

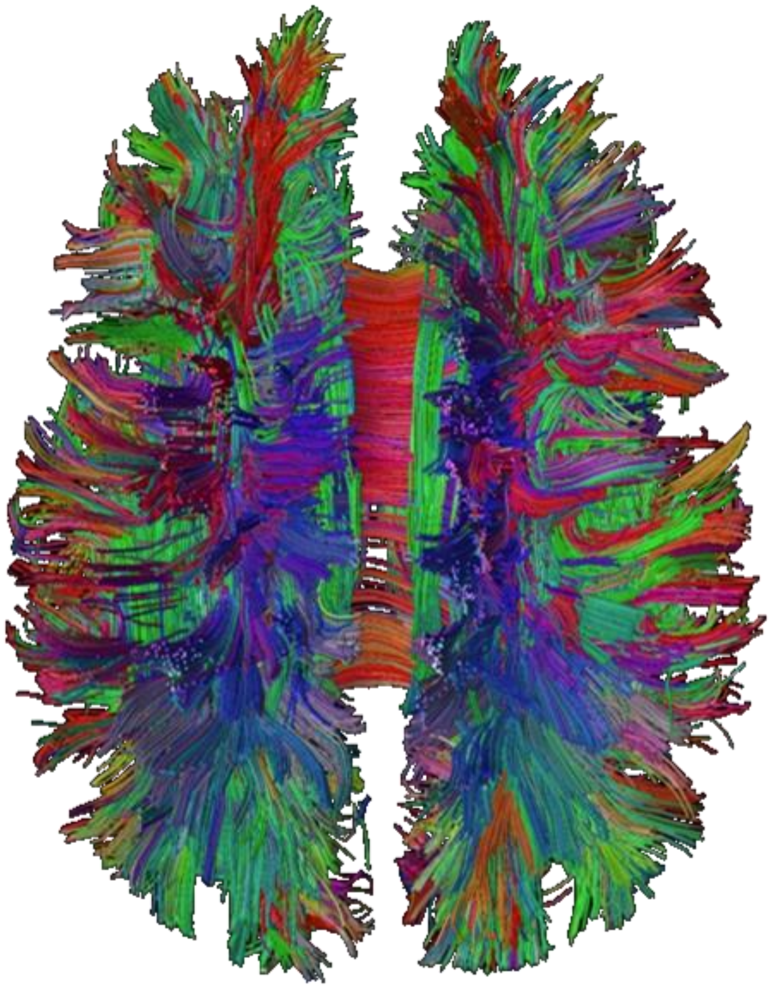
We need a neural network architecture that can handle:

- Data order
- Temporal dependencies
- Variable input sizes



Recurrent Neural Network

Brain is Highly Recurrent

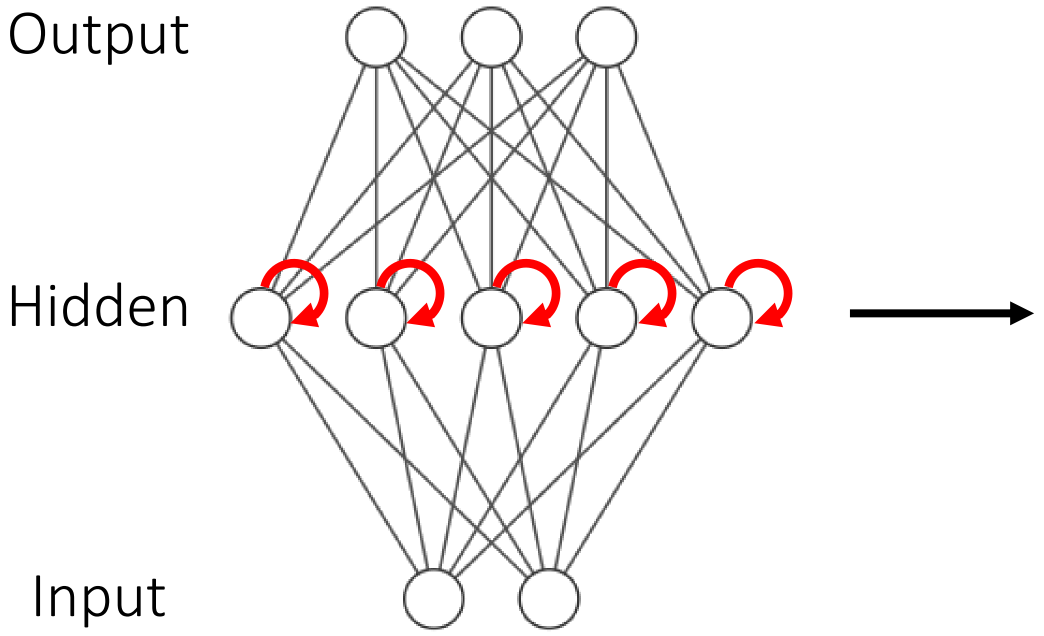


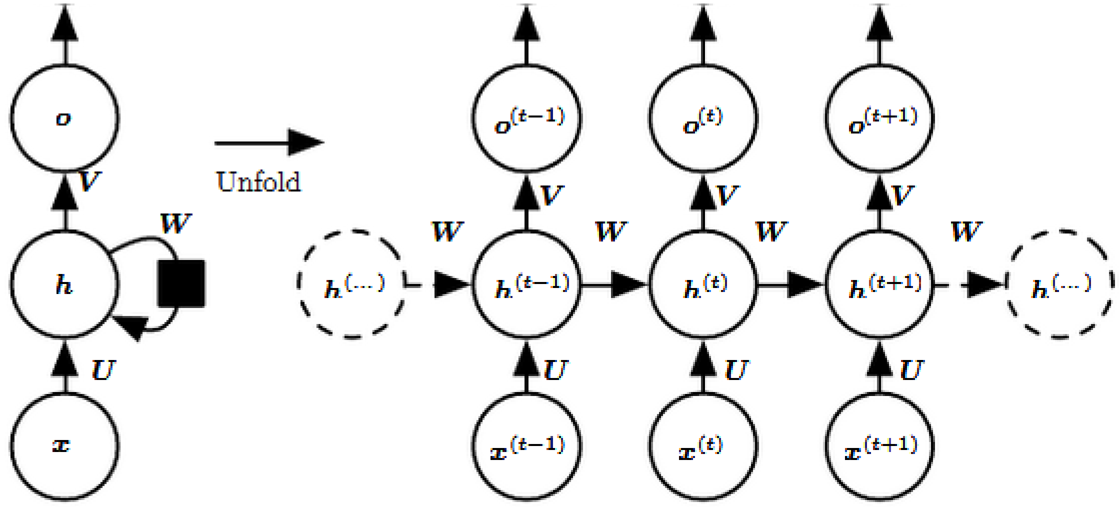
Neurons themselves have continuous voltage dynamics

Different parts of brain exchange information both forward and backward

Brain is Highly Recurrent

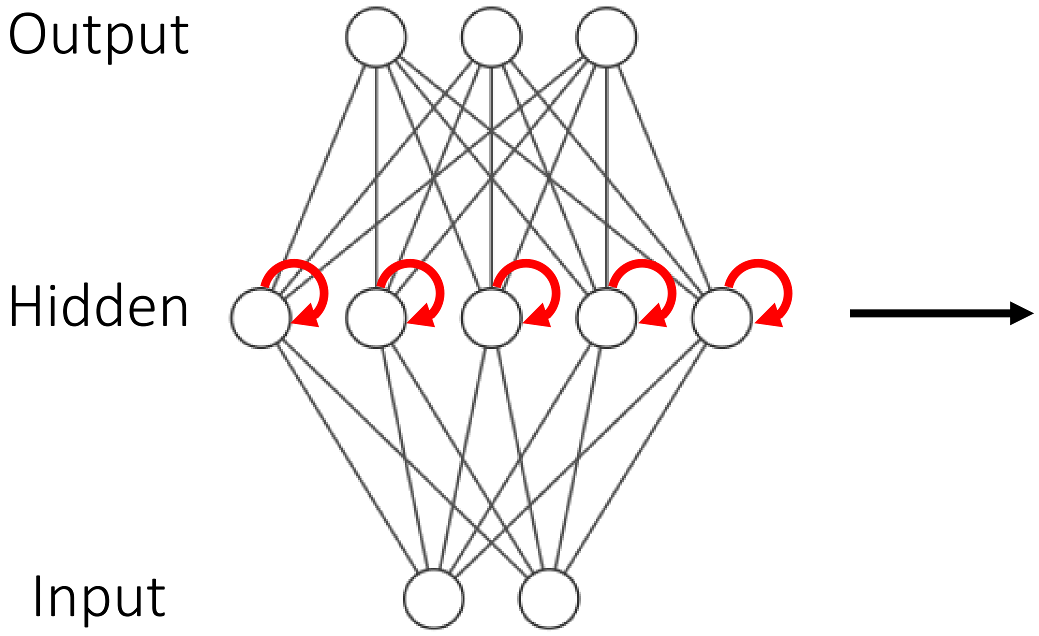
RNN Architecture

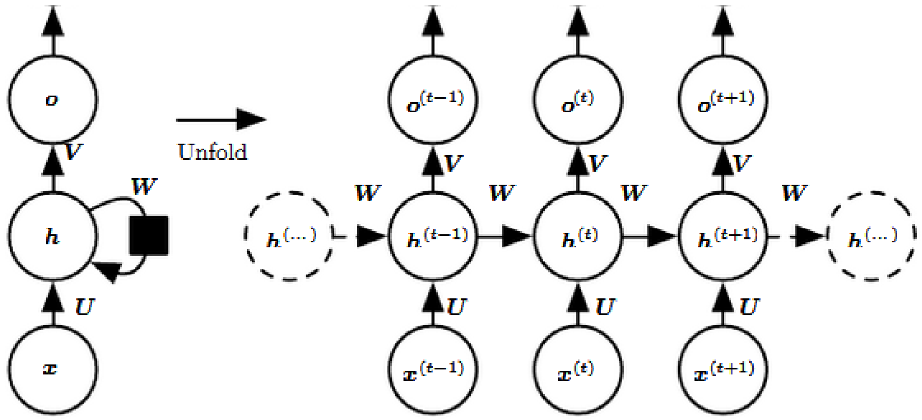




Unfold in Time

RNN Architecture

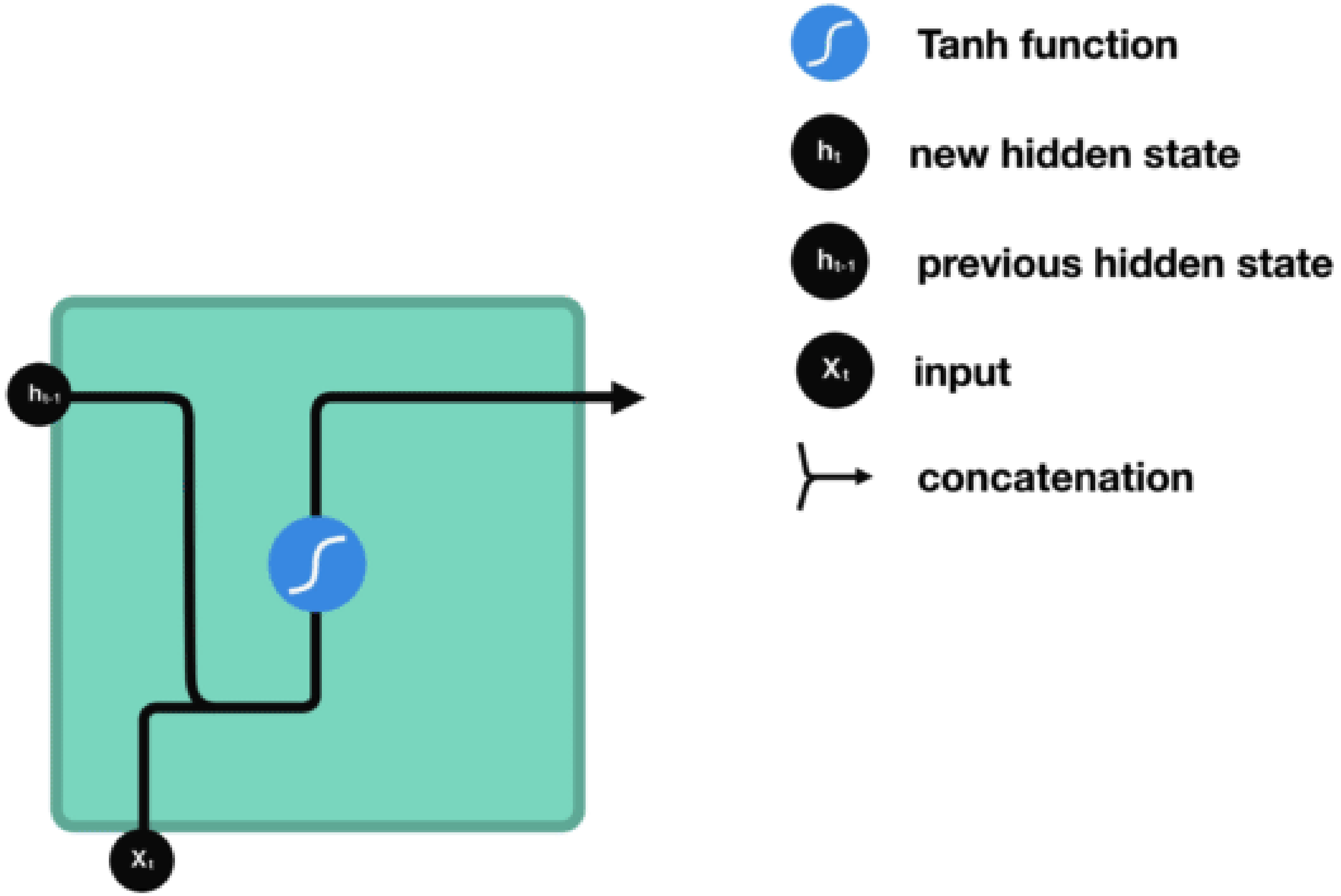




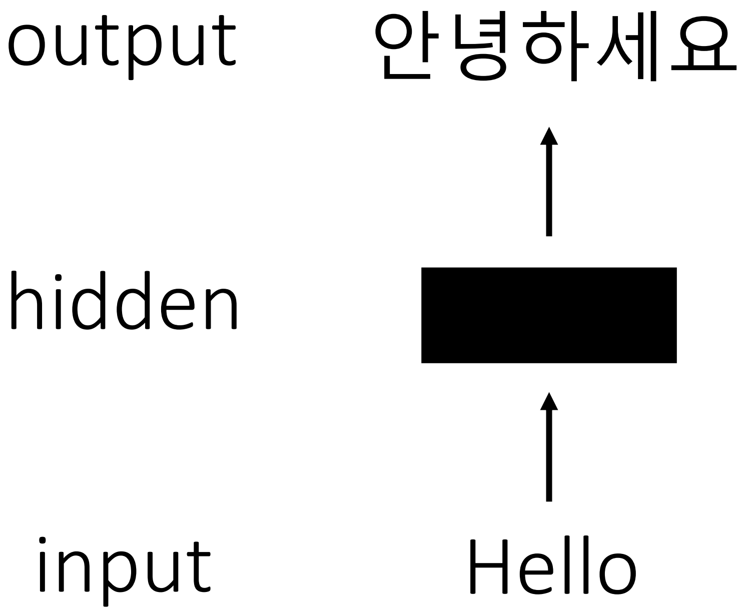
Unfold in Time

$$
\begin{aligned}
\boldsymbol{a}^{(t)} &= \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \\
\boldsymbol{h}^{(t)} &= \tanh(\boldsymbol{a}^{(t)}) \\
\boldsymbol{o}^{(t)} &= \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \\
\hat{\boldsymbol{y}}^{(t)} &= \operatorname{softmax}(\boldsymbol{o}^{(t)})
\end{aligned}
$$

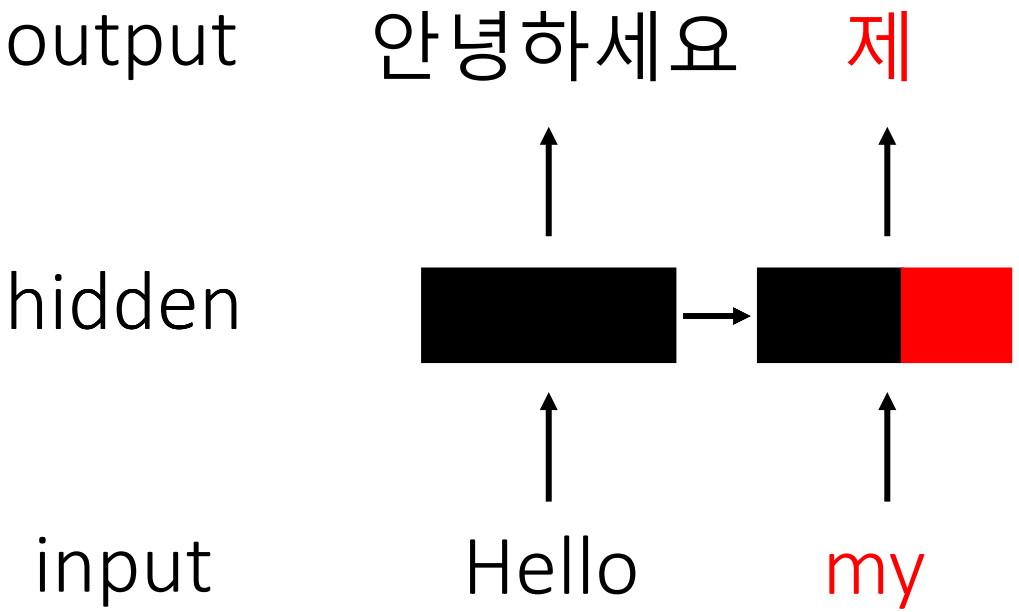
RNN Architecture



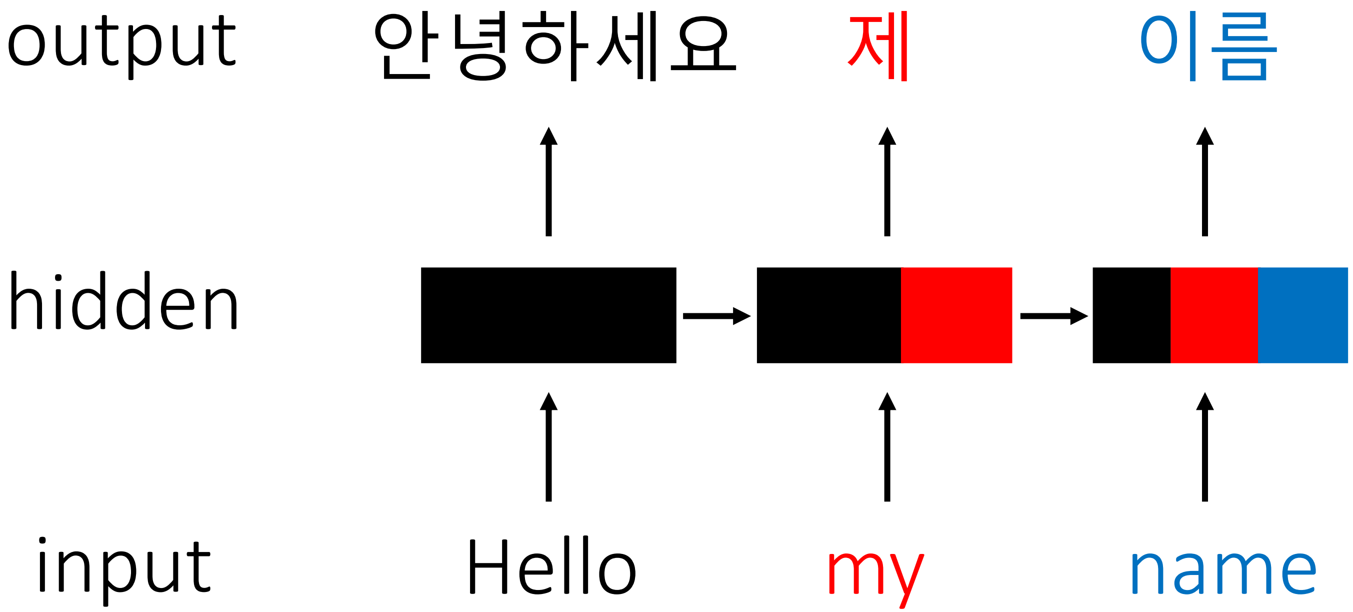
RNN Architecture



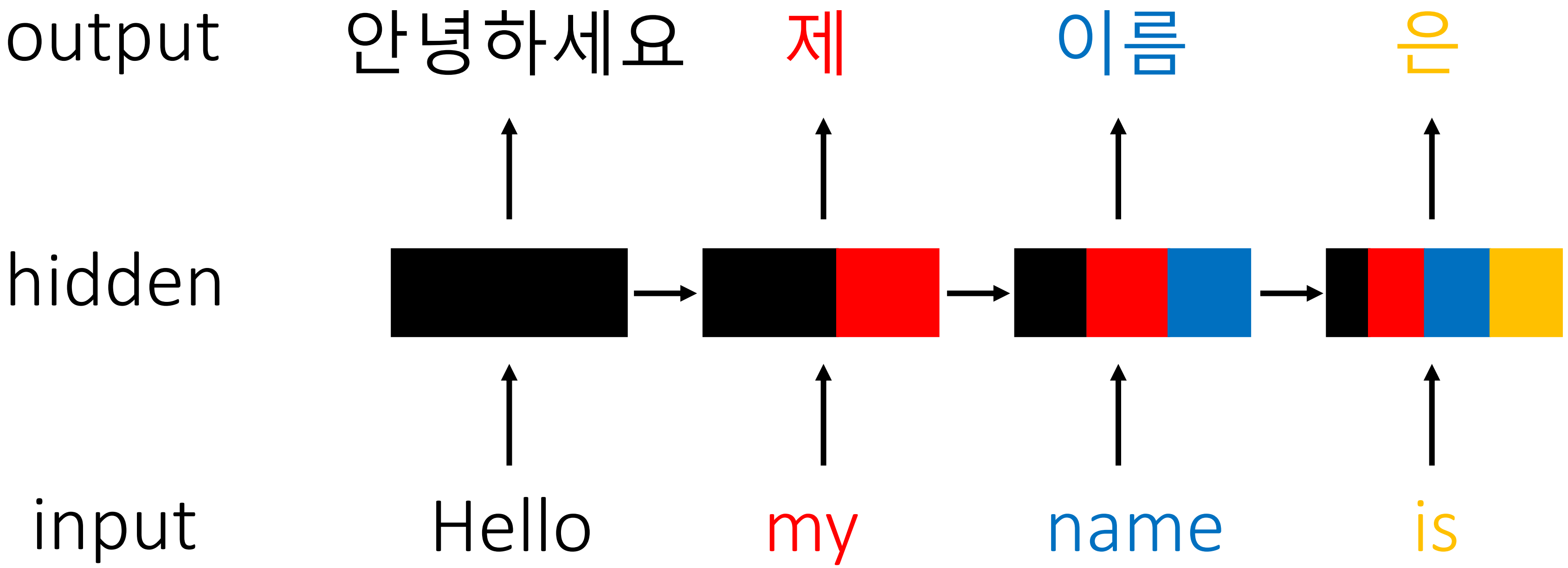
RNN Architecture



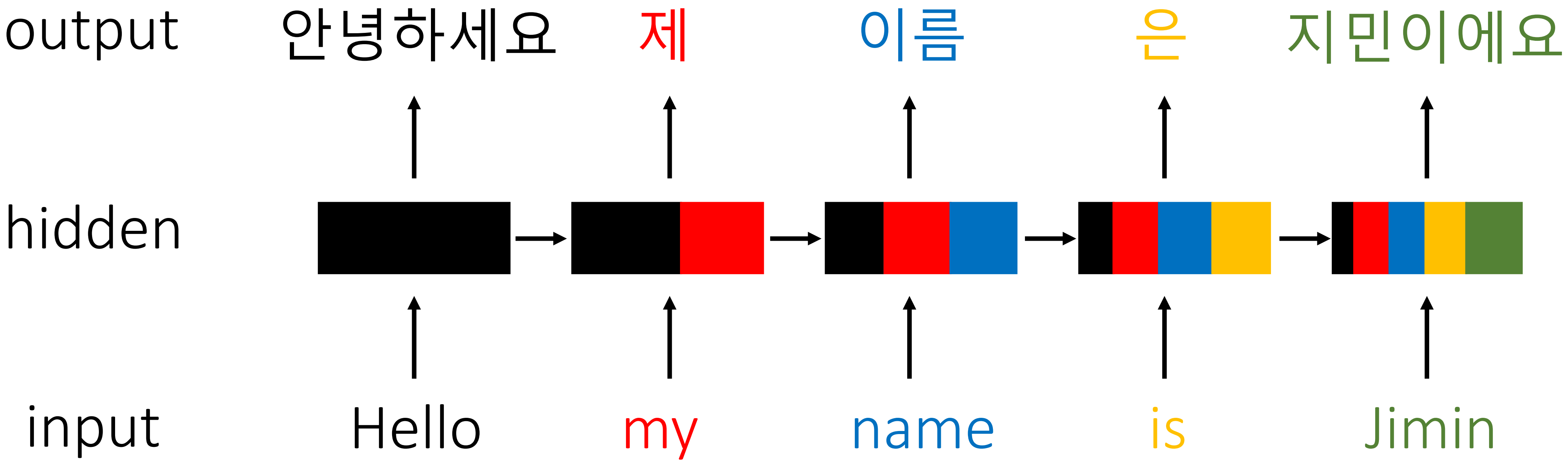
RNN Architecture



RNN Architecture



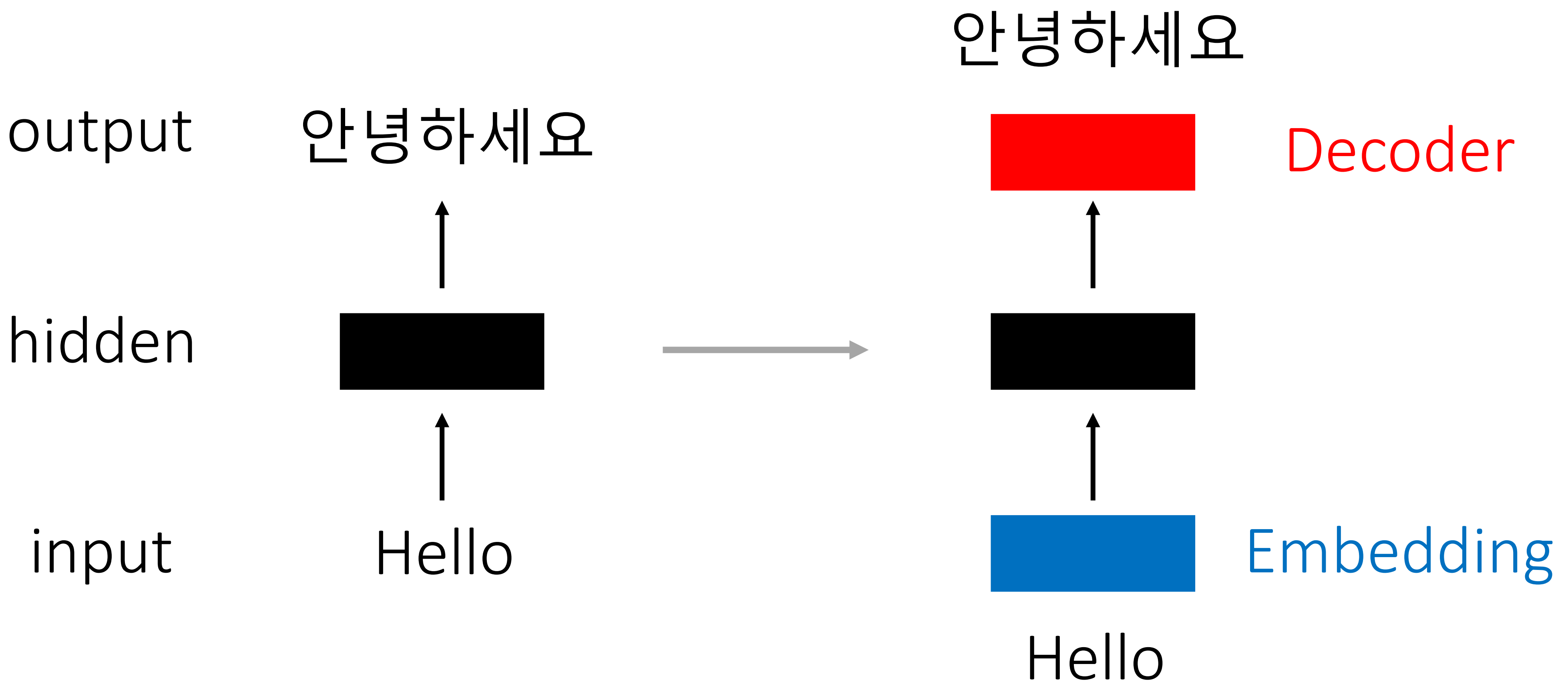
RNN Architecture



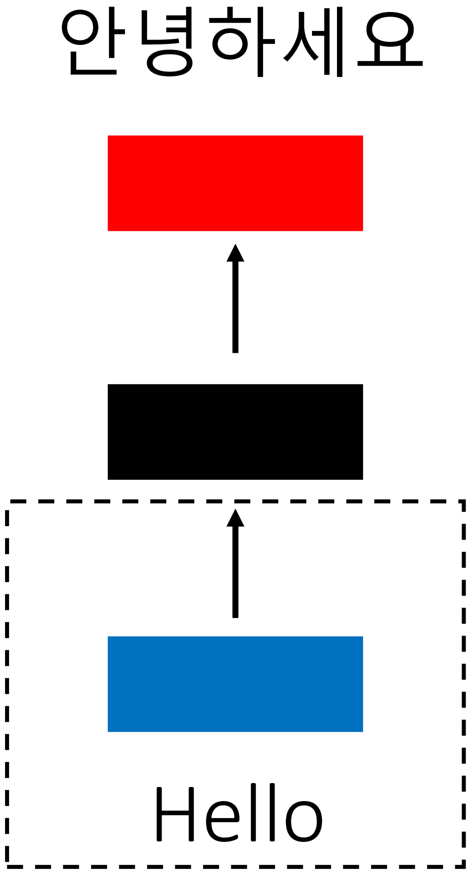
Sequential Data

Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

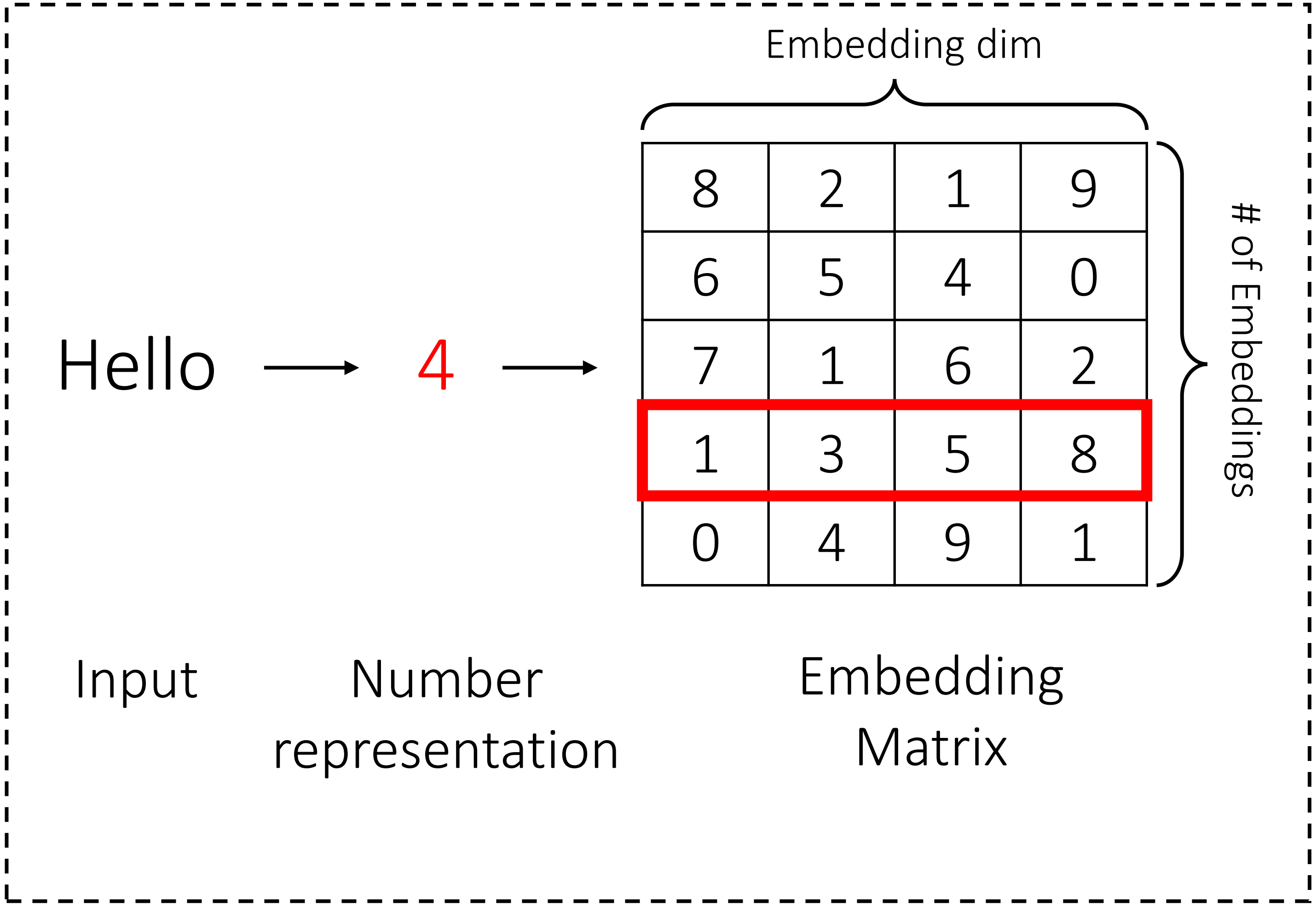
Embedding and Decoder



Embedding

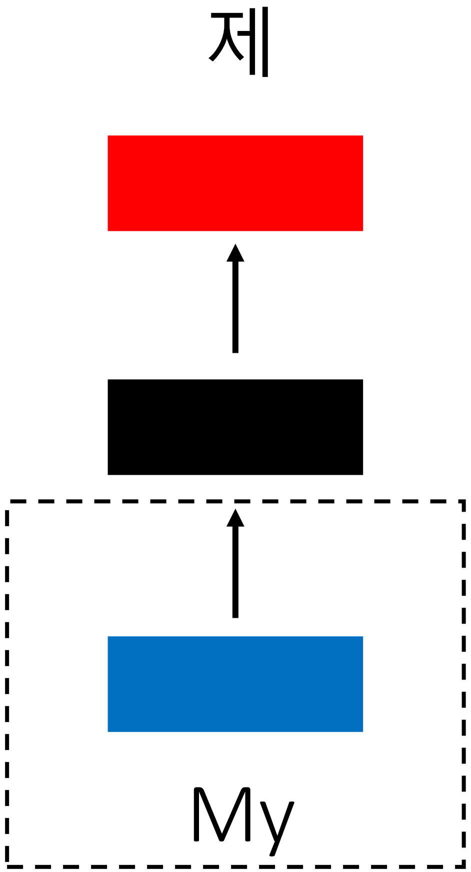


안녕하세요
Hello



Embedding dim
# of Embeddings
Hello
4
8 2 1 9
6 5 4 0
7 1 6 2
1 3 5 8
0 4 9 1
Input
Number representation
Embedding Matrix

Embedding

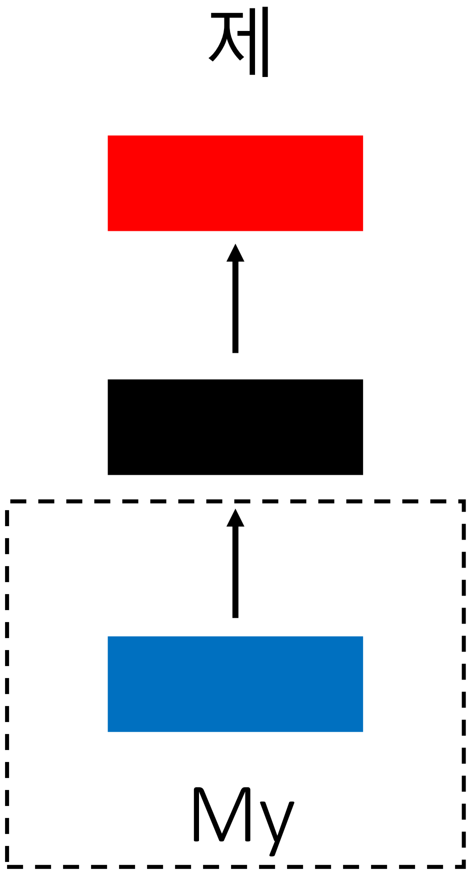


My → 2 →

8	2	1	9
6	5	4	0
7	1	6	2
1	3	5	8
0	4	9	1

Input — Number representation — Embedding Matrix

Embedding



My → 2 →

8	2	1	9
6	5	4	0
7	1	6	2
1	3	5	8
0	4	9	1

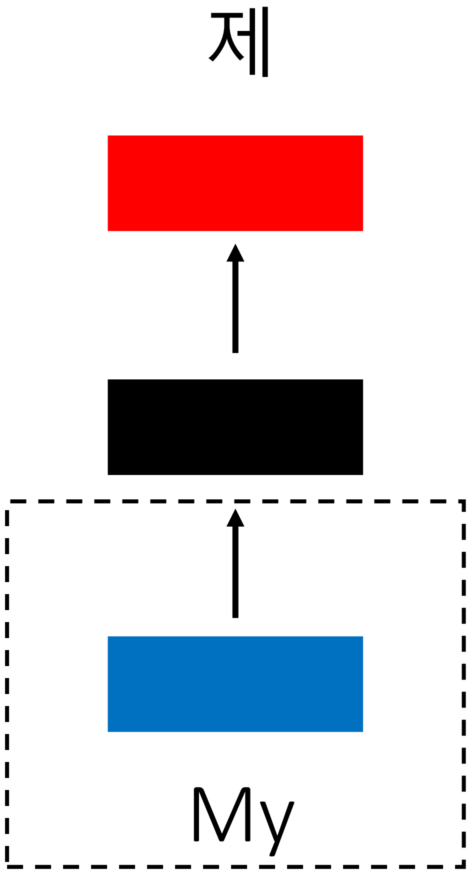
Input

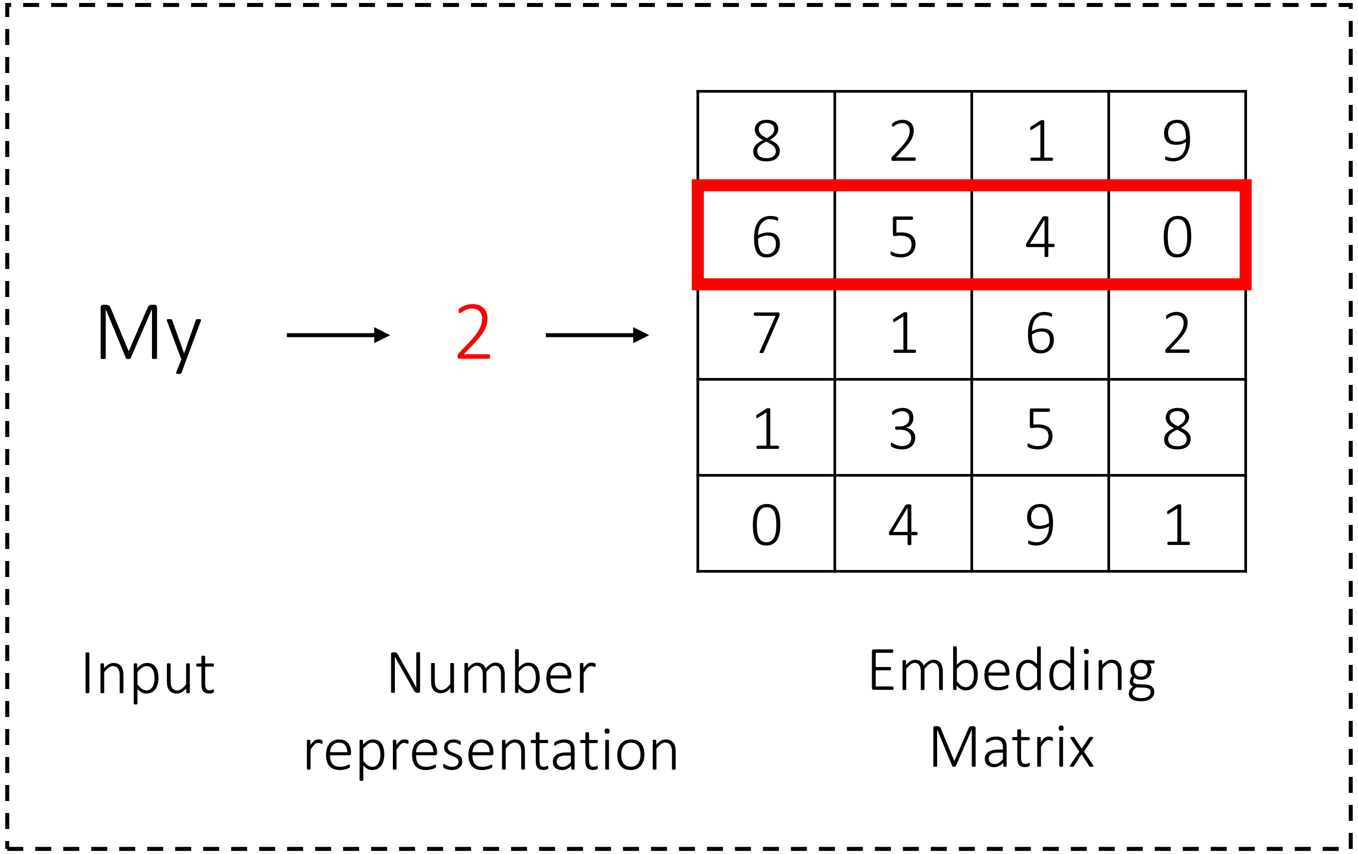
Number representation

Embedding Matrix

Embedding matrix is trainable

Embedding

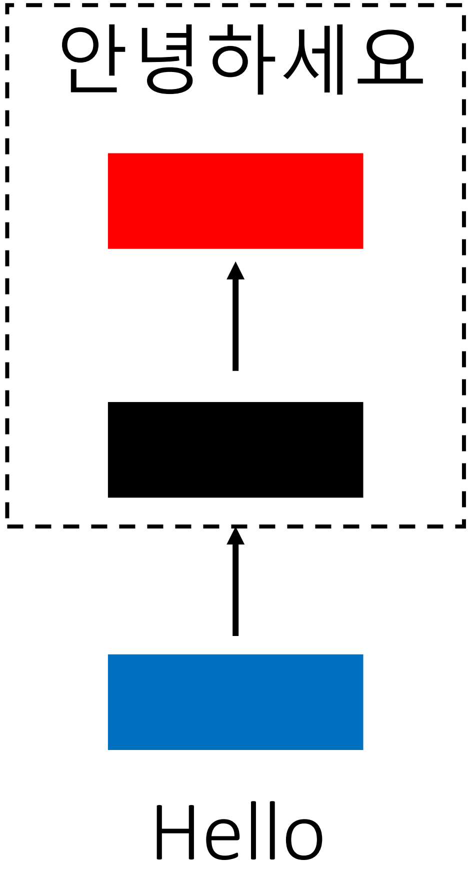


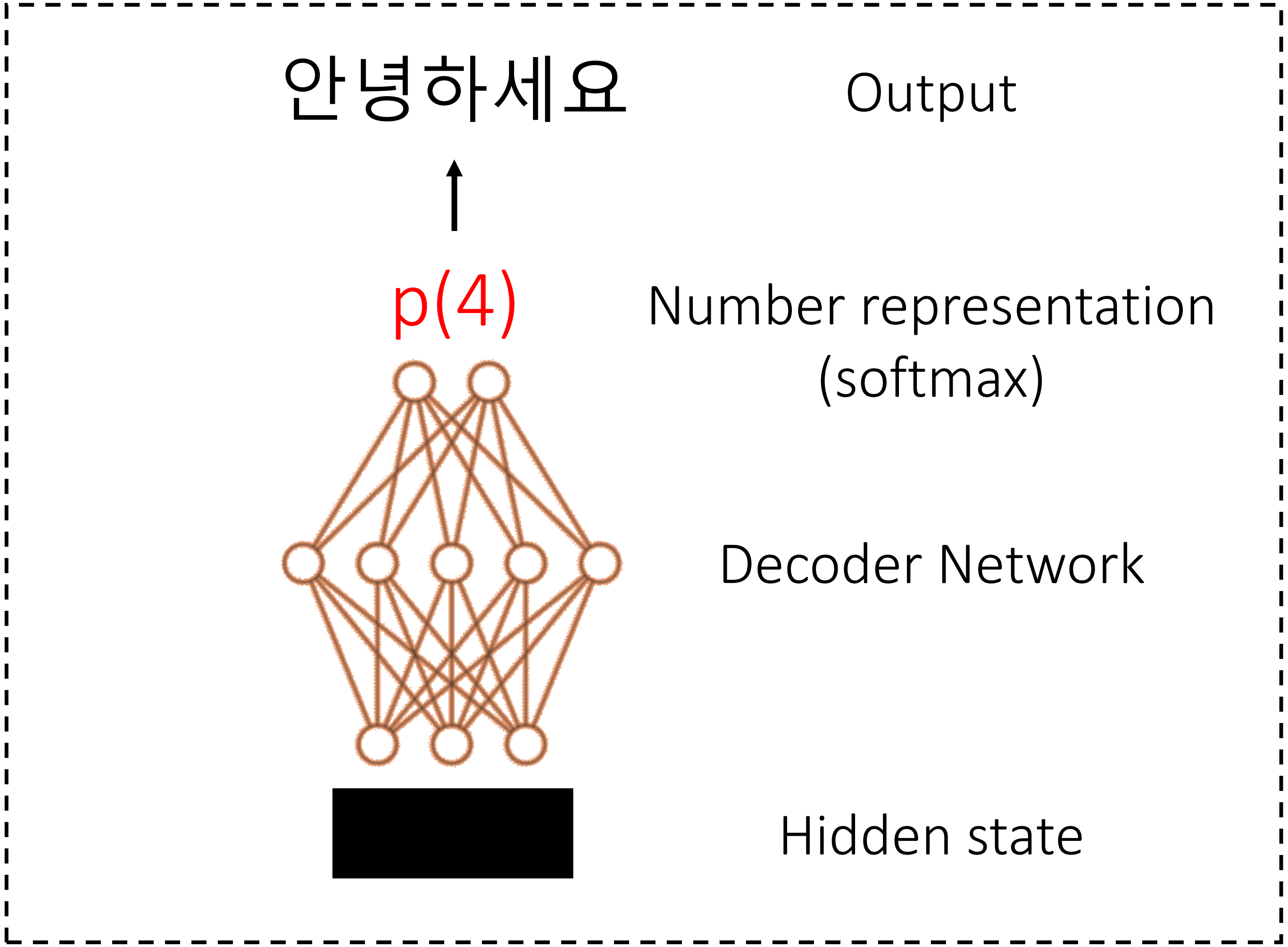


torch.nn.embedding(num_embeddings, embedding dim)

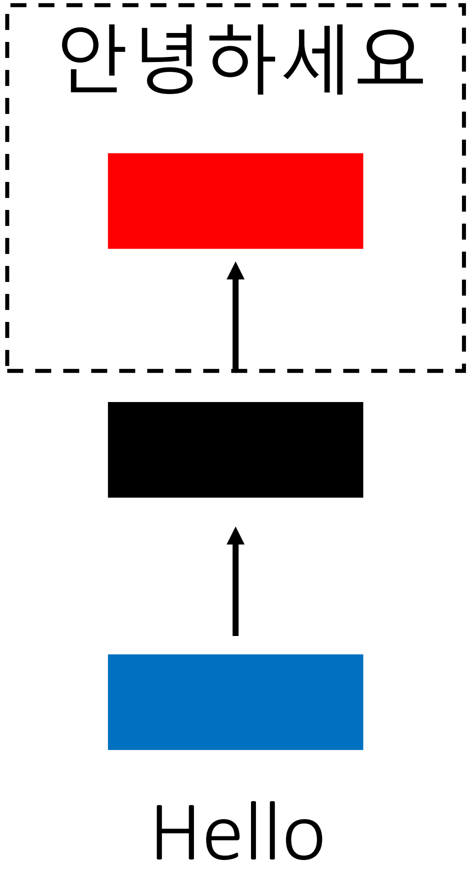
https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html

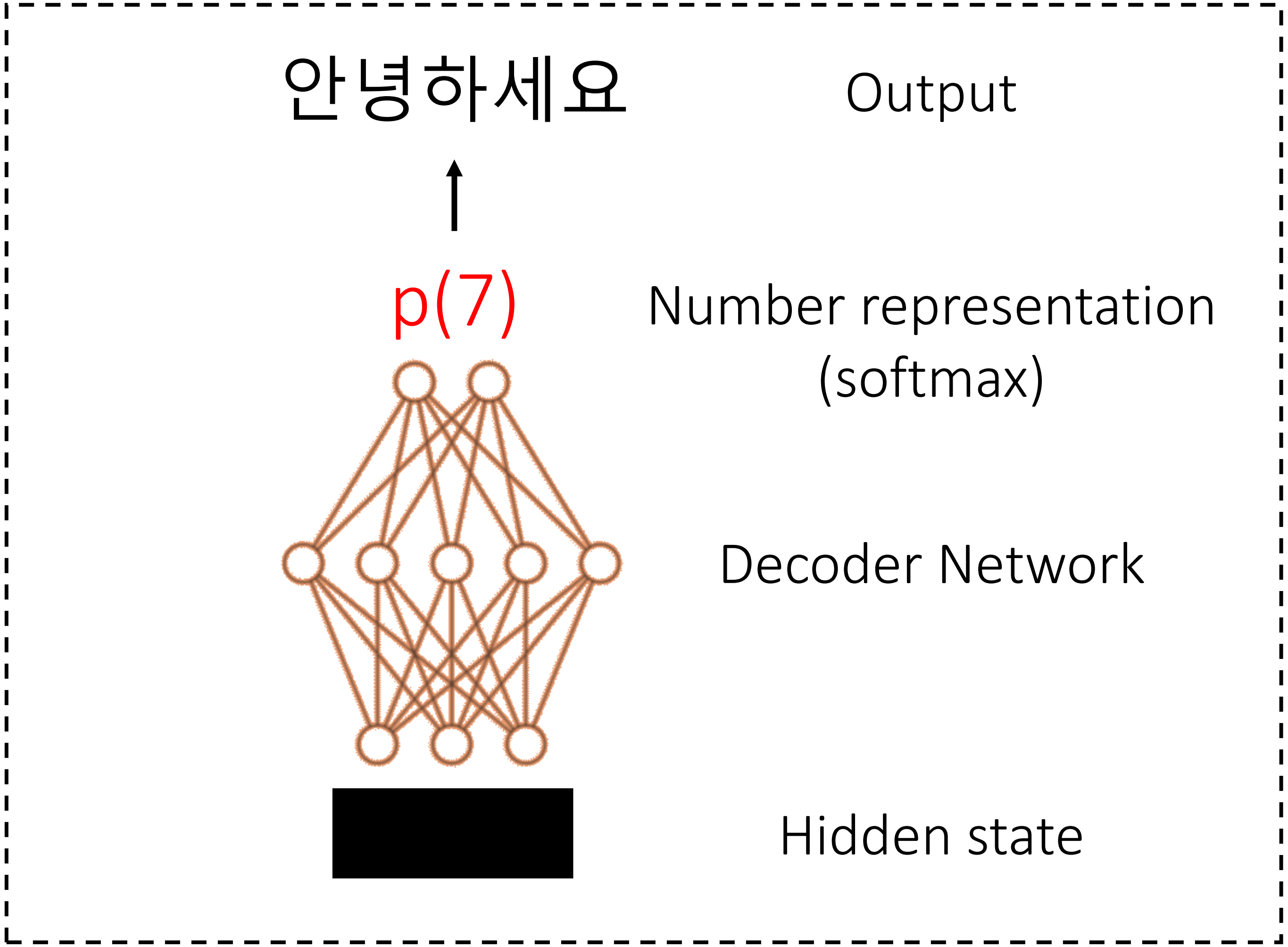
Decoder


안녕하세요
Hello


안녕하세요
Output
p(4)
Number representation
(softmax)
Decoder Network
Hidden state

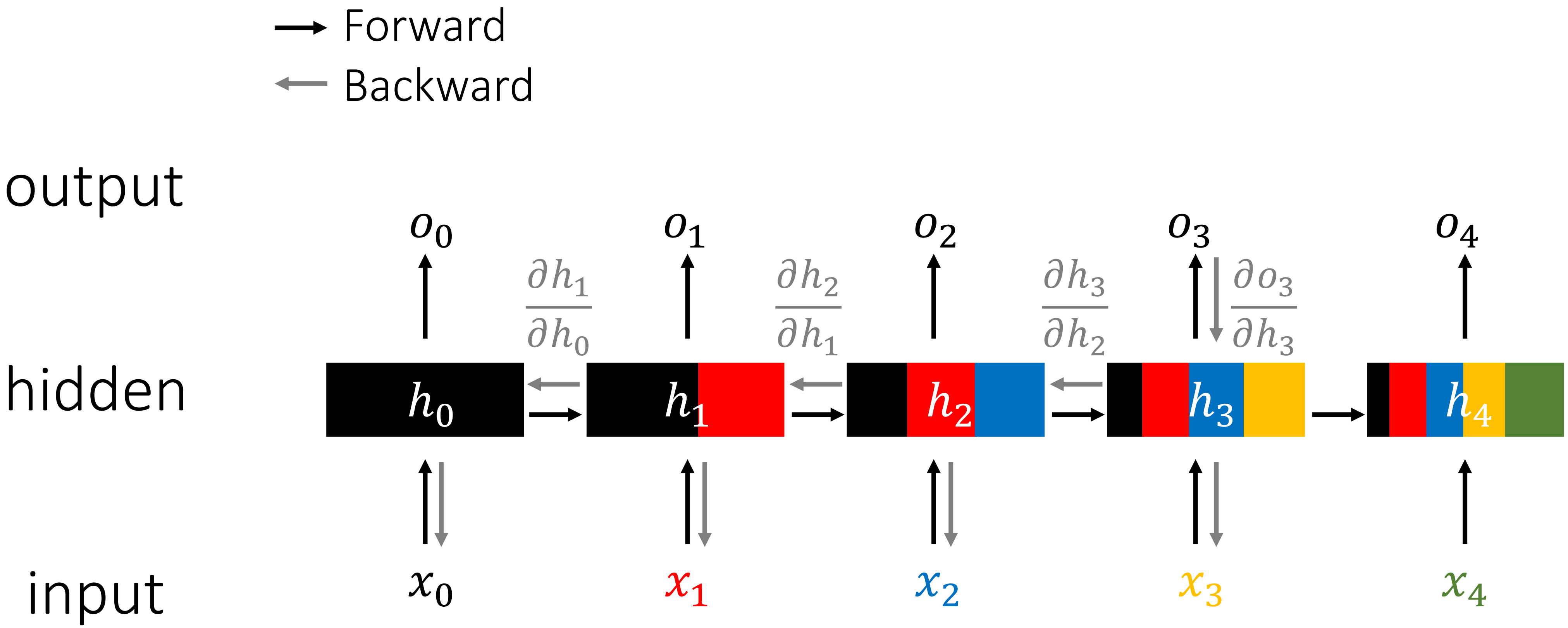
Decoder



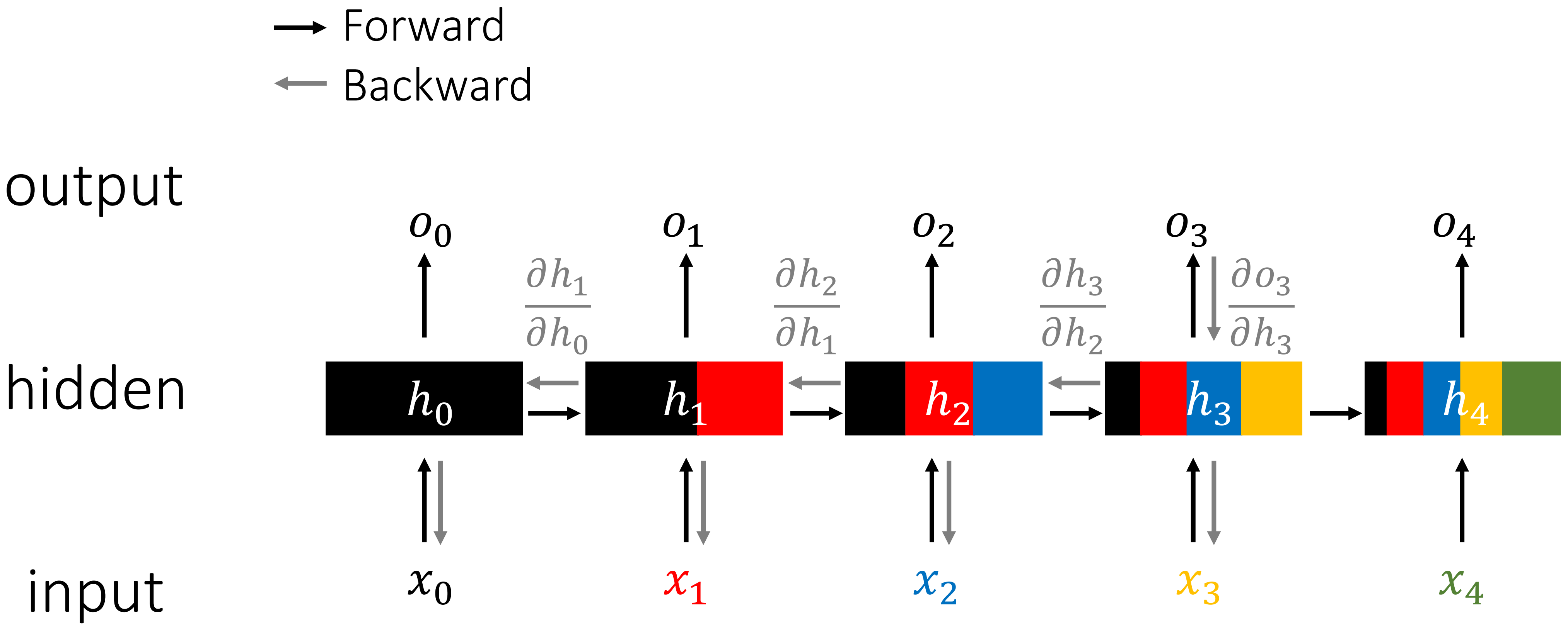


torch.nn.Linear(hidden_size, output_size)

Backpropagation in RNNs

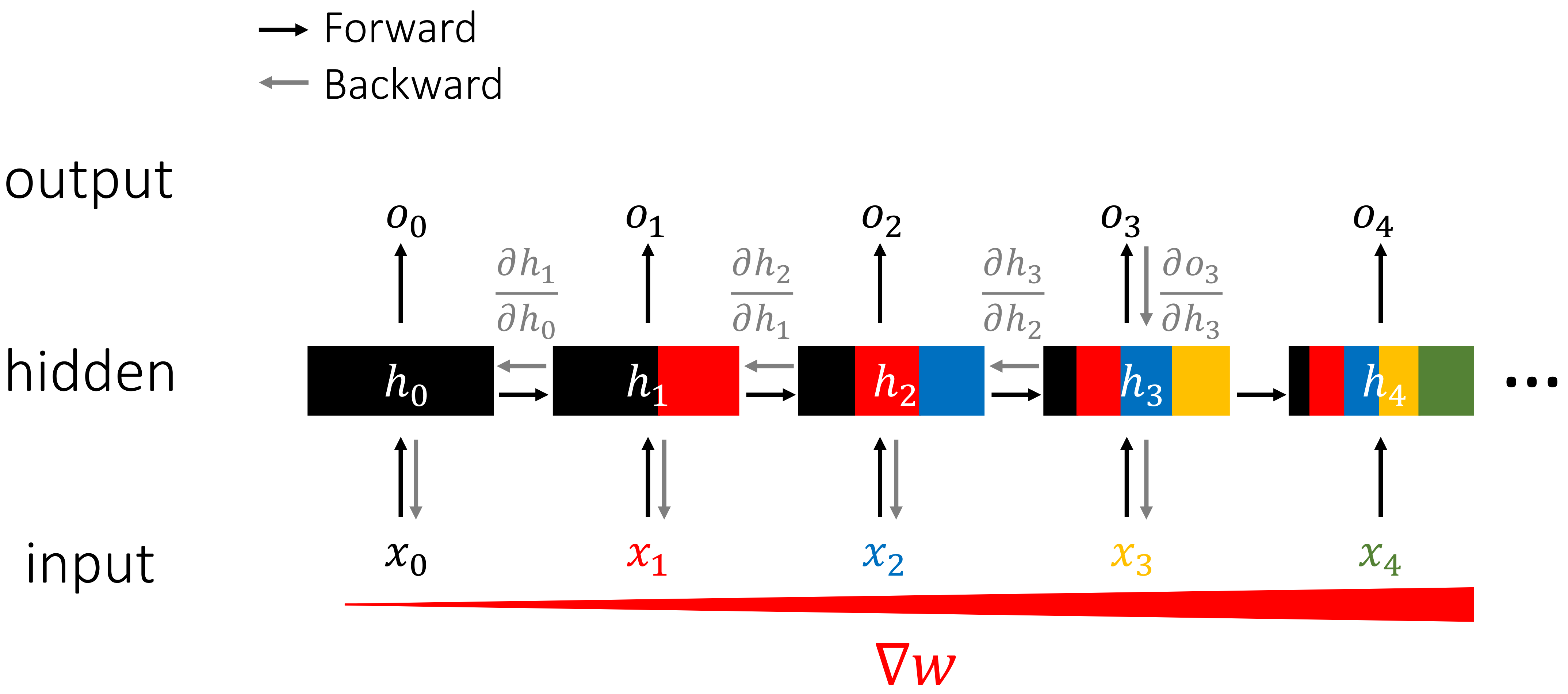


Backpropagation in RNNs

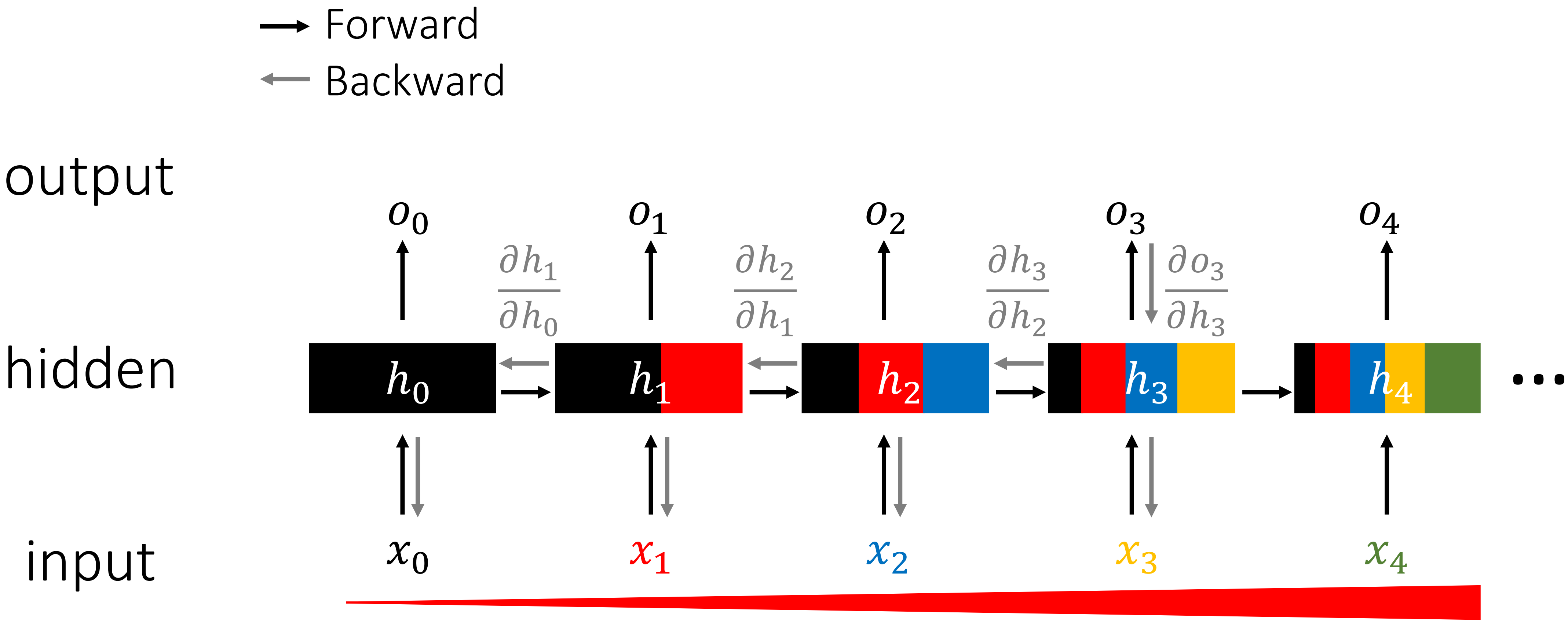


Backpropagation is performed backward in time

Vanishing and Exploding Gradients



Vanishing and Exploding Gradients



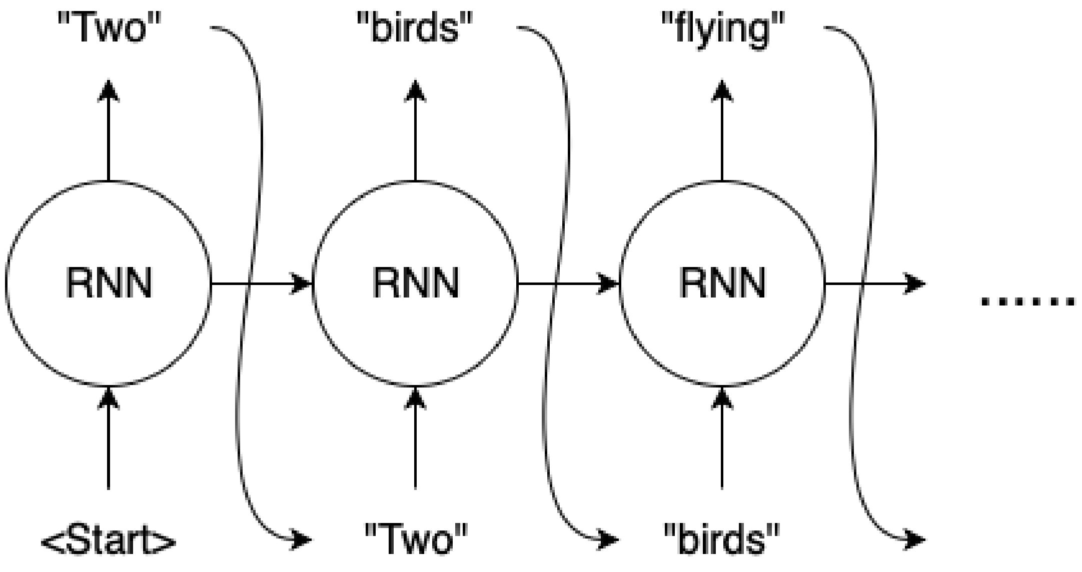
Longer input sequence →

higher risk of Vanishing/Exploding Gradients!

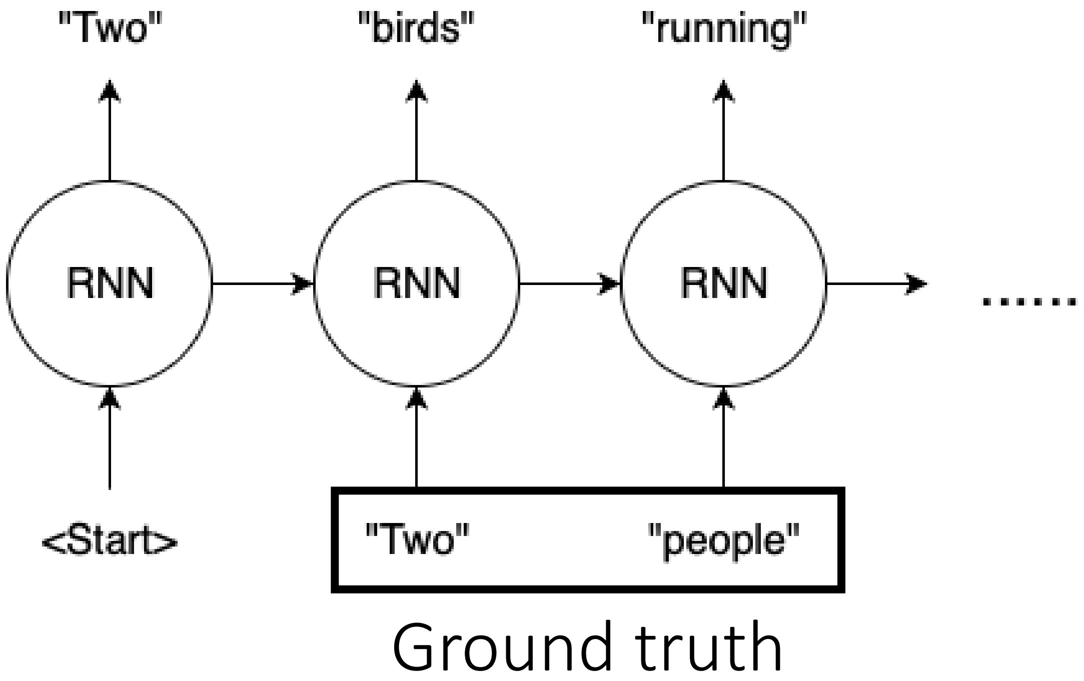
Vanishing and Exploding Gradients

- Use gated RNN architecture e.g., LSTM, GRU (Lab 6)
- ReLU activation as nonlinearity
- Smaller number of sequence
- Smaller learning rate

Training RNN with Teacher Forcing



Without Teacher Forcing



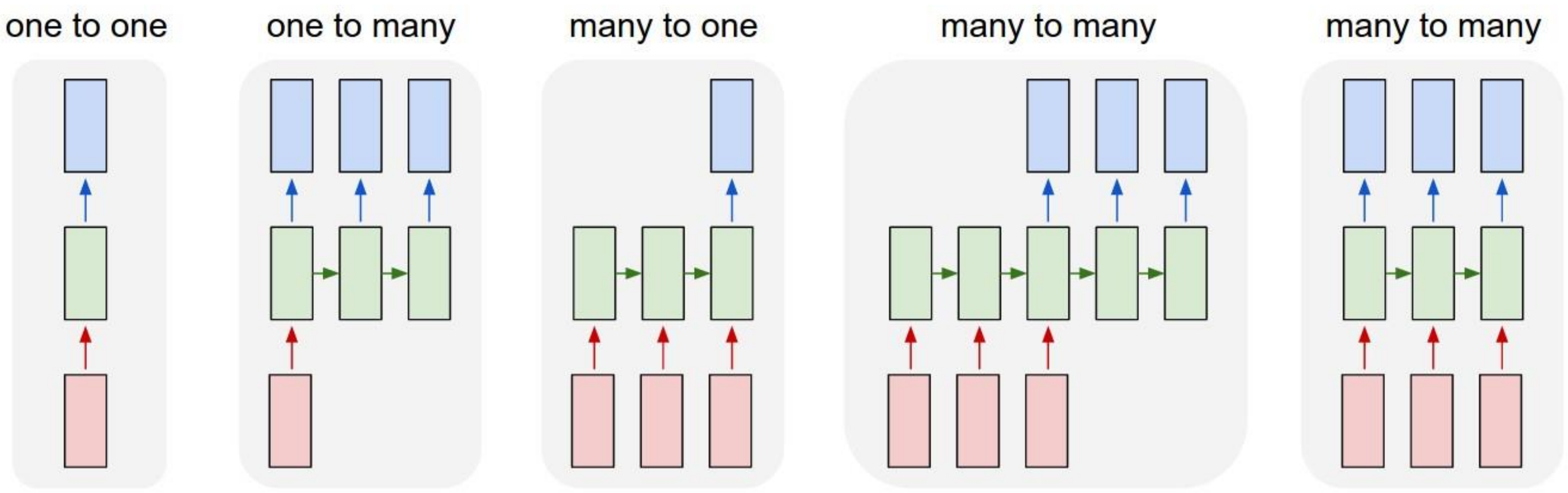
With Teacher Forcing

RNN PROBLEM TYPES

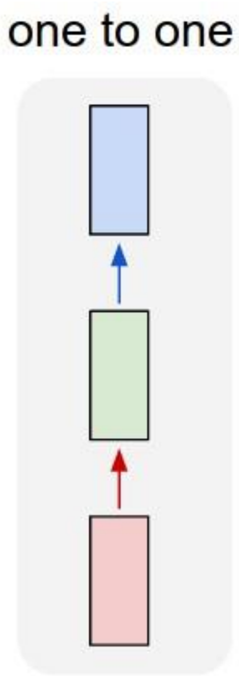
RNN Configurations

RNN Extensions

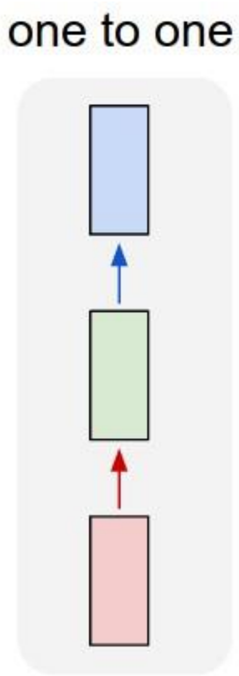
RNN Configurations



One to One

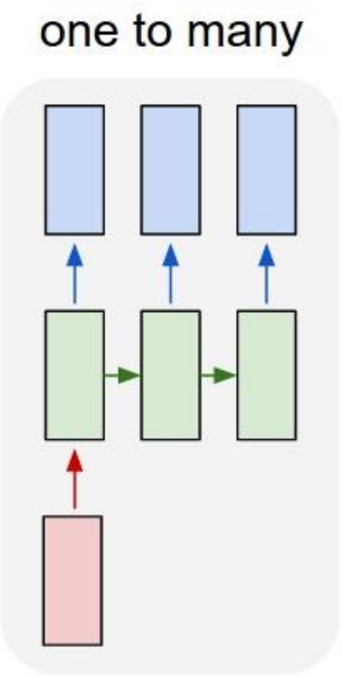


One to One



Identical to Feed Forward Network

One to Many



One to Many

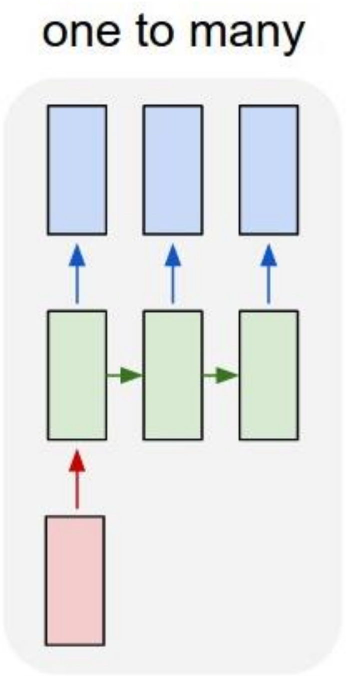
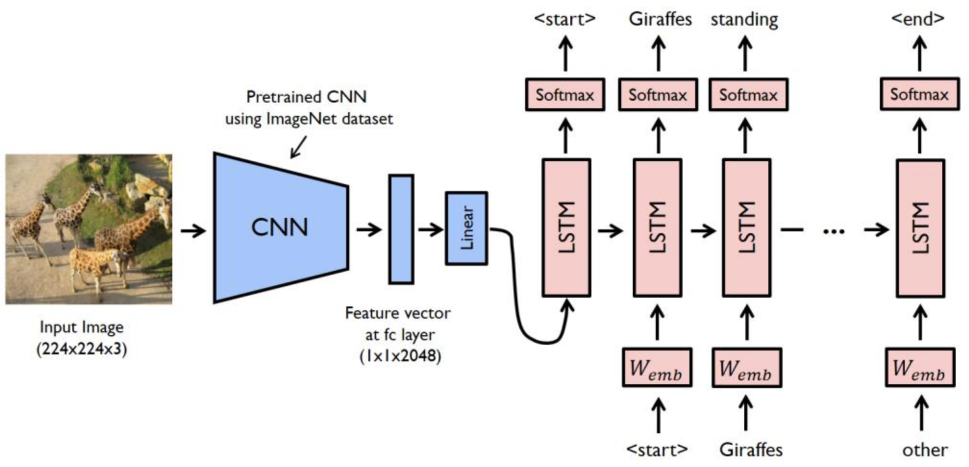
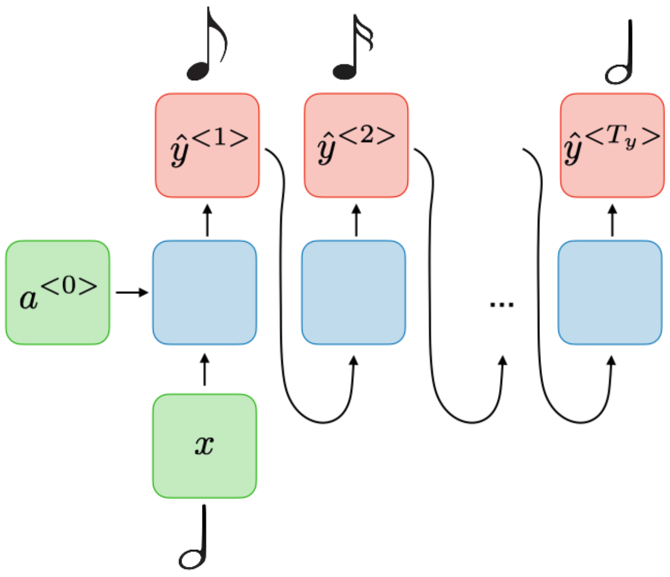


Image credit: www.analyticsvidhya.com

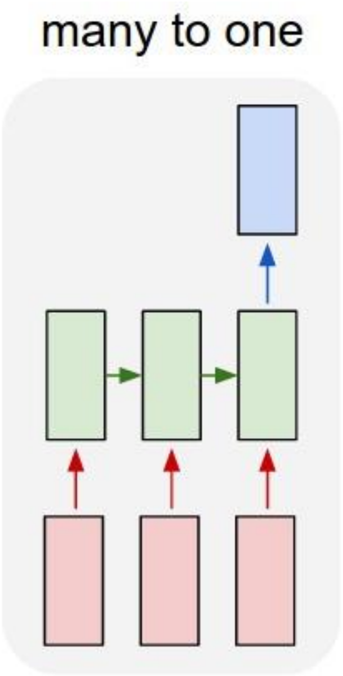
Image captioning



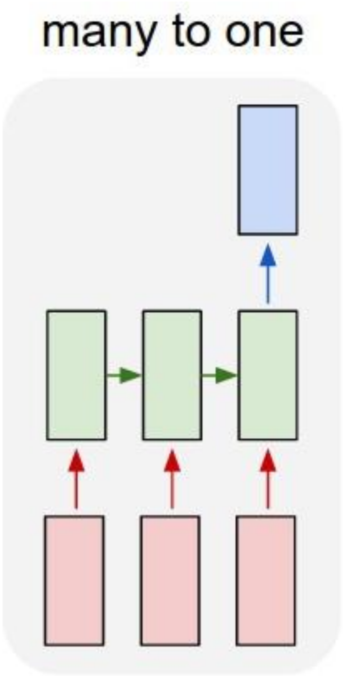
Music generation

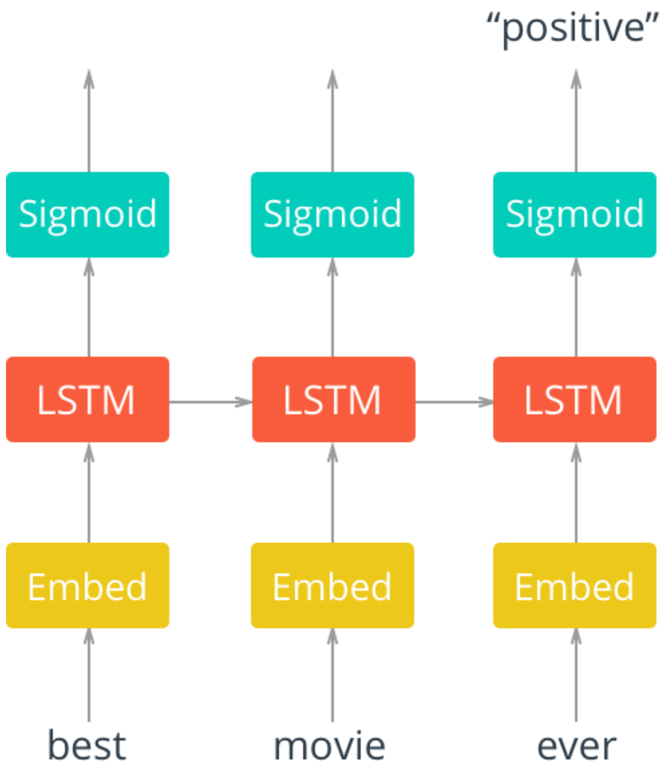


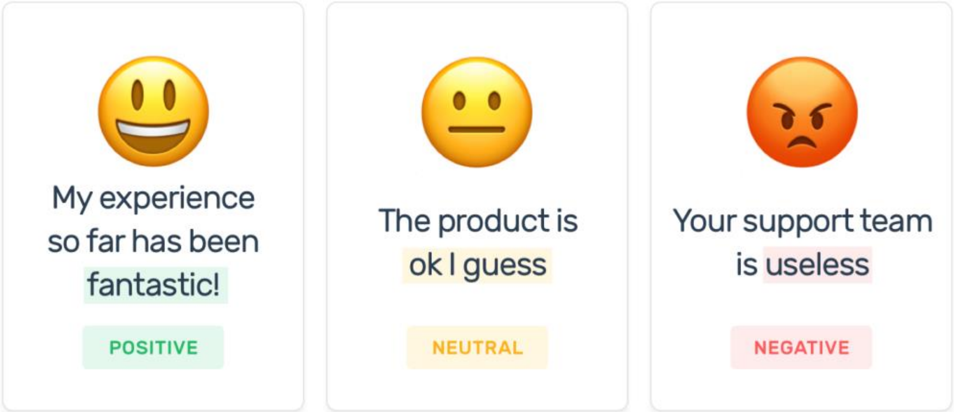
Many to One



Many to One

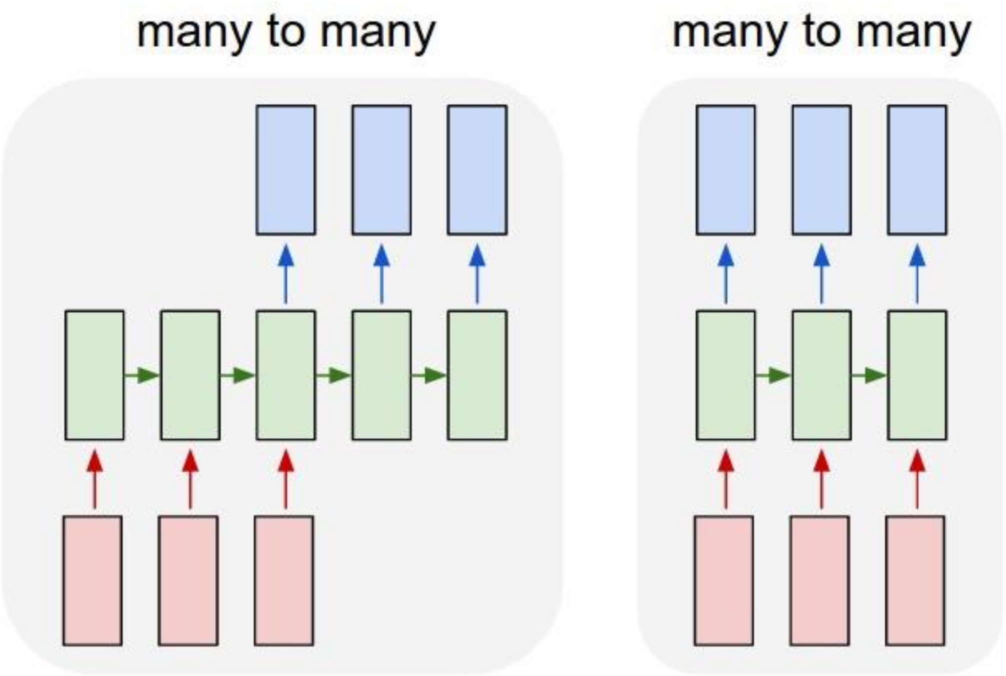




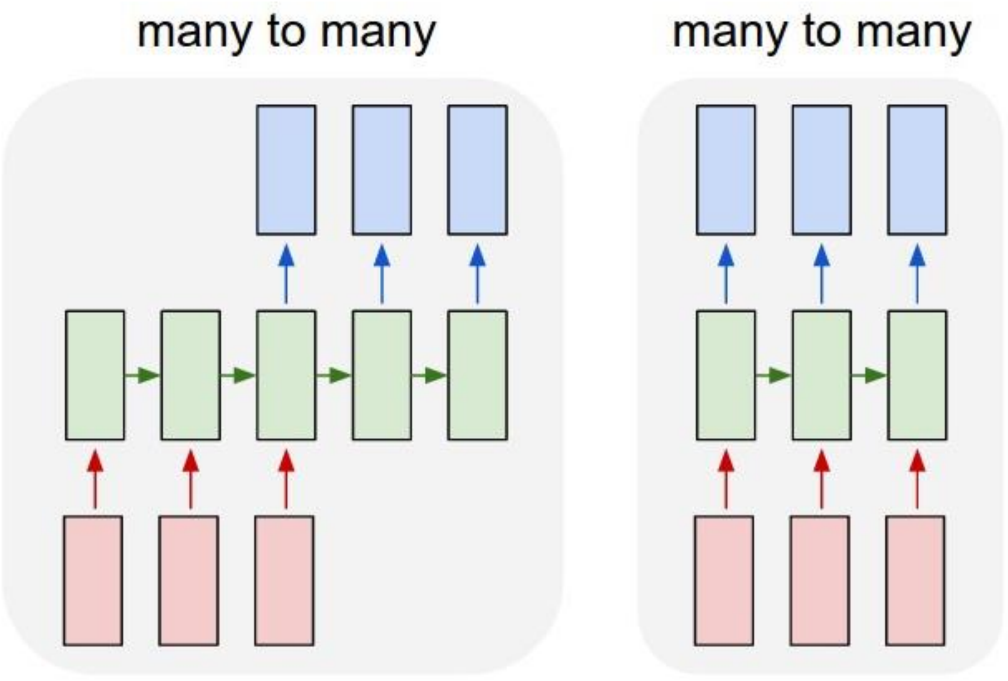


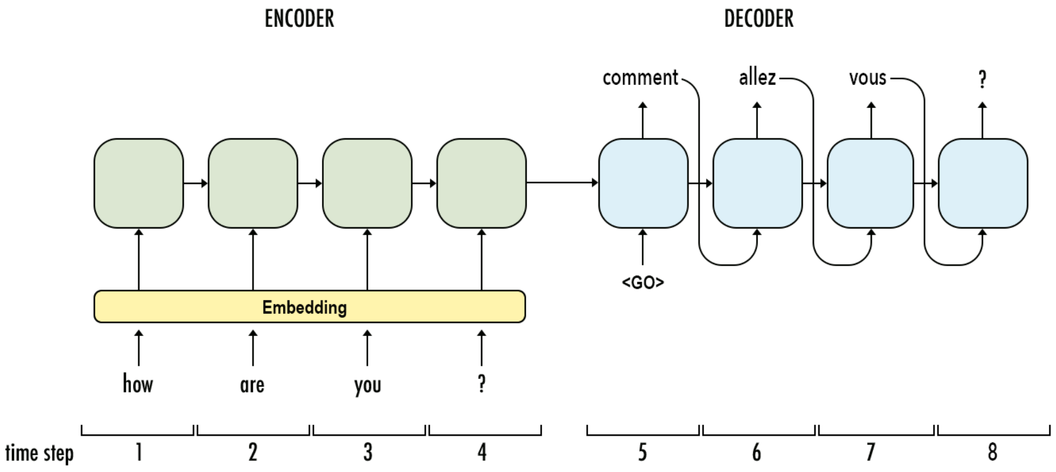
Sentiment Analysis

Many to Many

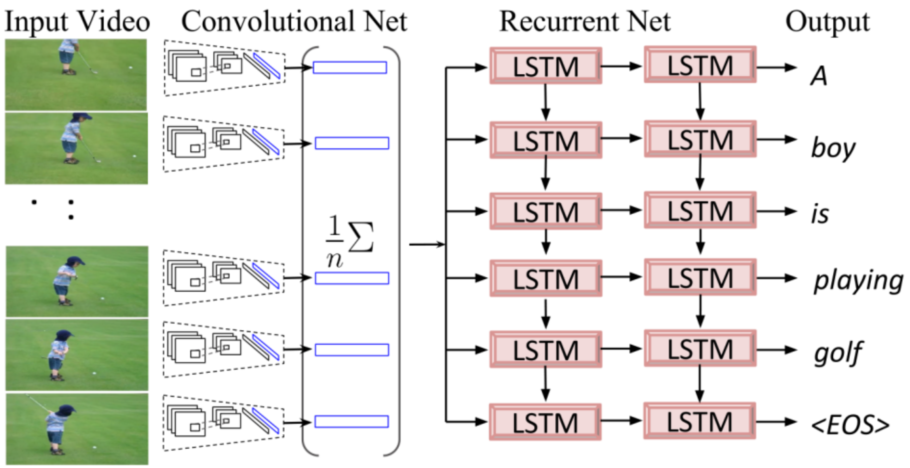


Many to Many



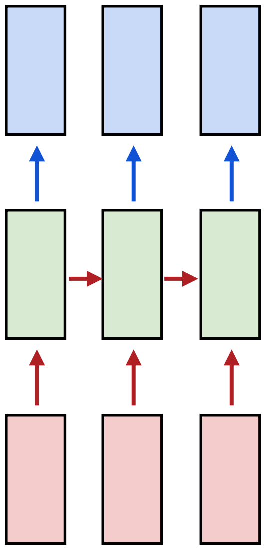


Machine Translation

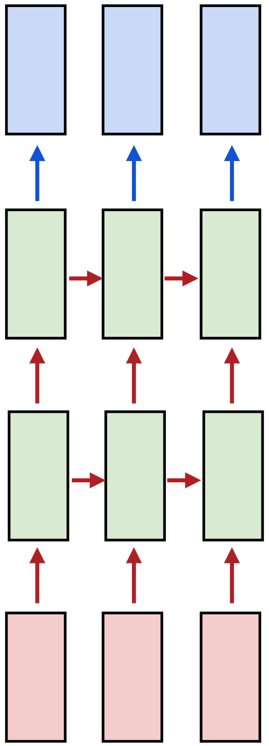


Video Captioning

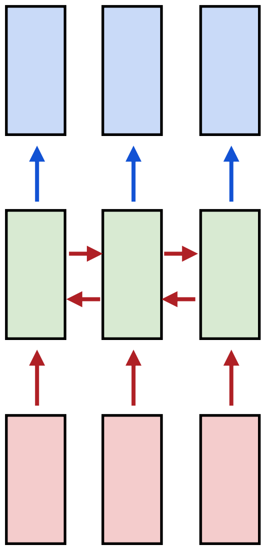
RNN Extensions



Regular RNN

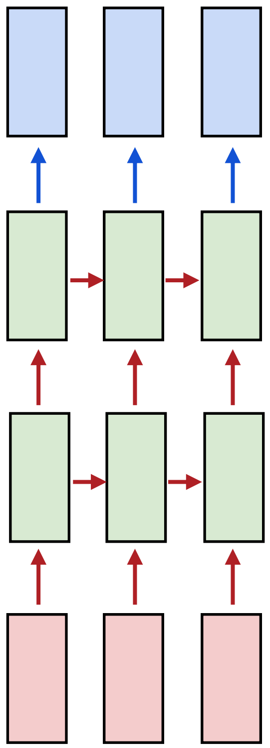


Deep RNN



Bi-directional RNN

Deep RNN



Deep RNN

(+)

Can provide better performance

Often used for complex problems

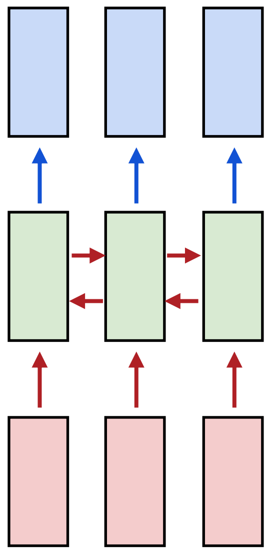
(-)

Potential for overfitting

Longer training time

Implemented as ‘num_layers’ parameters in torch.nn.RNN()

Bi-directional RNNs



Bi-directional RNN

(+)

Higher performance in Natural Language Processing tasks

Suitable when both left and right contexts are used

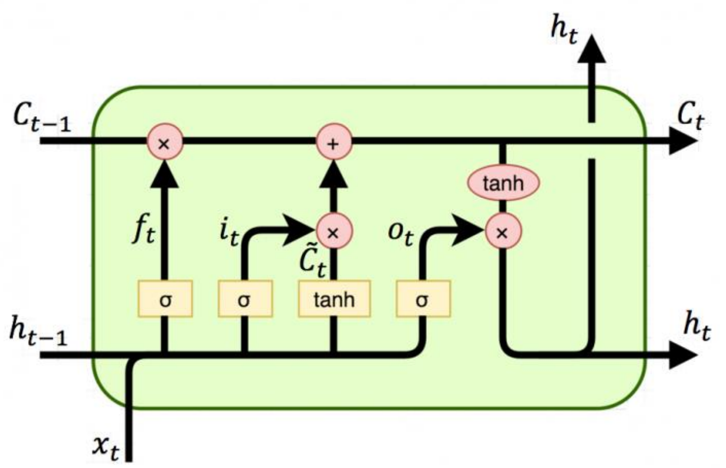
(-)

Harder to train than Uni-directional RNN

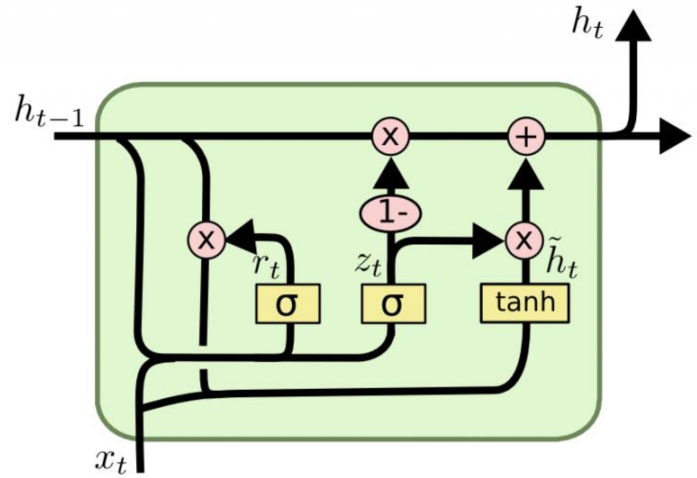
Not suitable for real-time processing

Implemented as ‘bidirectional’ parameter in torch.nn.RNN()

Next episode in EEP 596…



Long-short-term memory
(LSTM)



Gated recurrent units
(GRU)

Next episode in EEP 596…