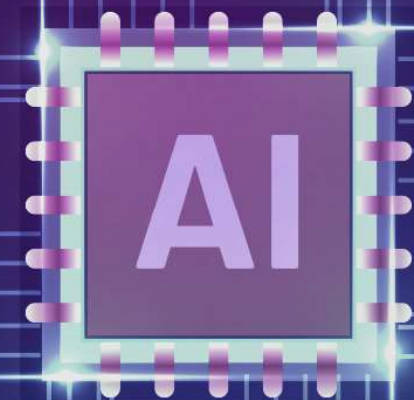


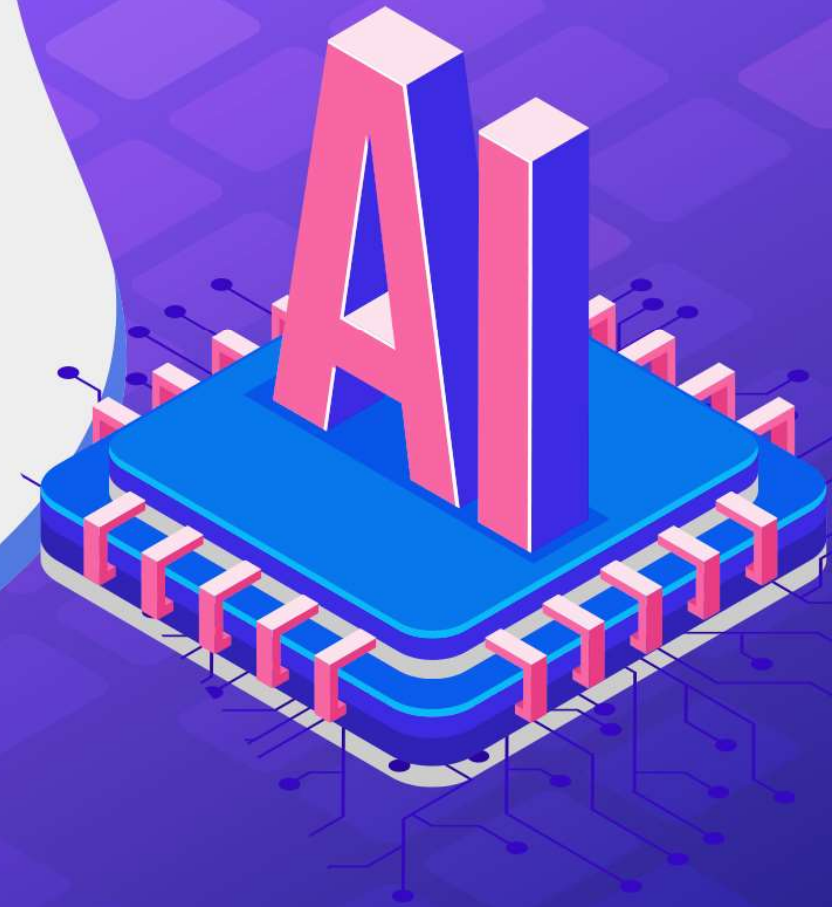
RNN (with Pytorch)

김재광 교수 (소프트웨어융합대학 글로벌융합학부)



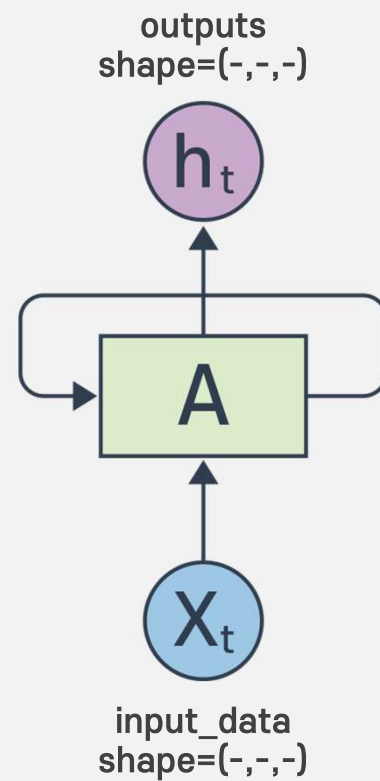
Contents

- RNN in PyTorch
- Simple Example
 - ✓ Input
 - ✓ Hidden State
 - ✓ Sequence Length
 - ✓ Batch Size



RNN in PyTorch

```
rnn = torch.nn.RNN(input_size, hidden_size)
outputs, _status = rnn(input_data)
```



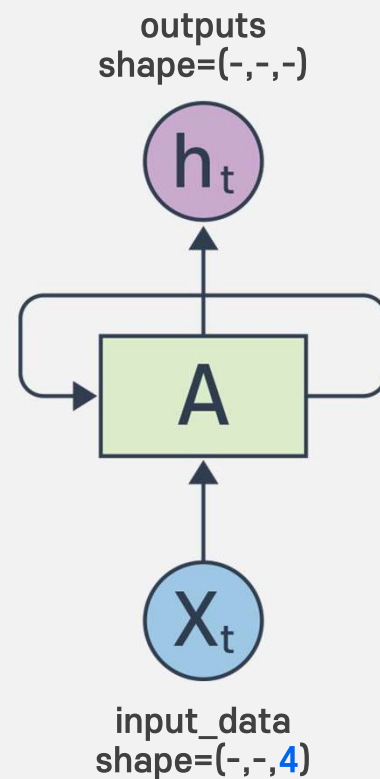
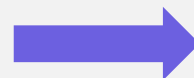
Simple Example

● Input

"hello"

```
# 1-hot encoding  
h = [1, 0, 0, 0]  
e = [0, 1, 0, 0]  
l = [0, 0, 1, 0]  
o = [0, 0, 0, 1]
```

input_size = 4



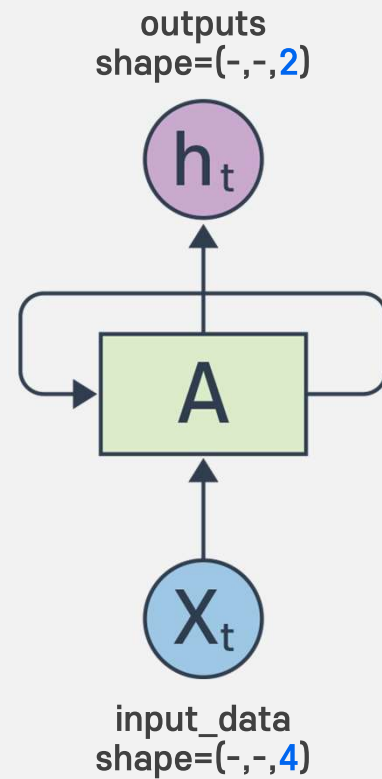
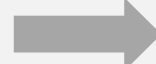
Simple Example (Cont'd)

- Hidden State

desirable output size

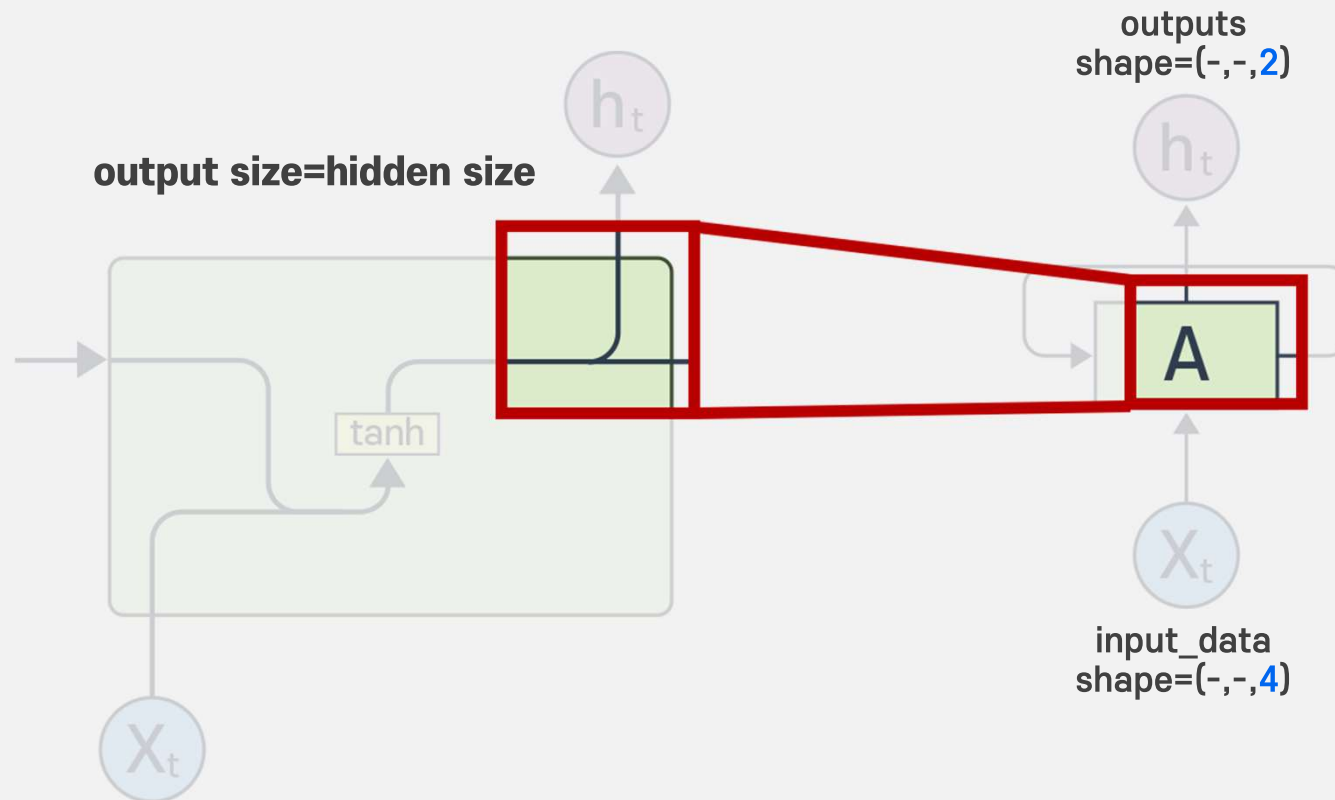


hidden_size = 2



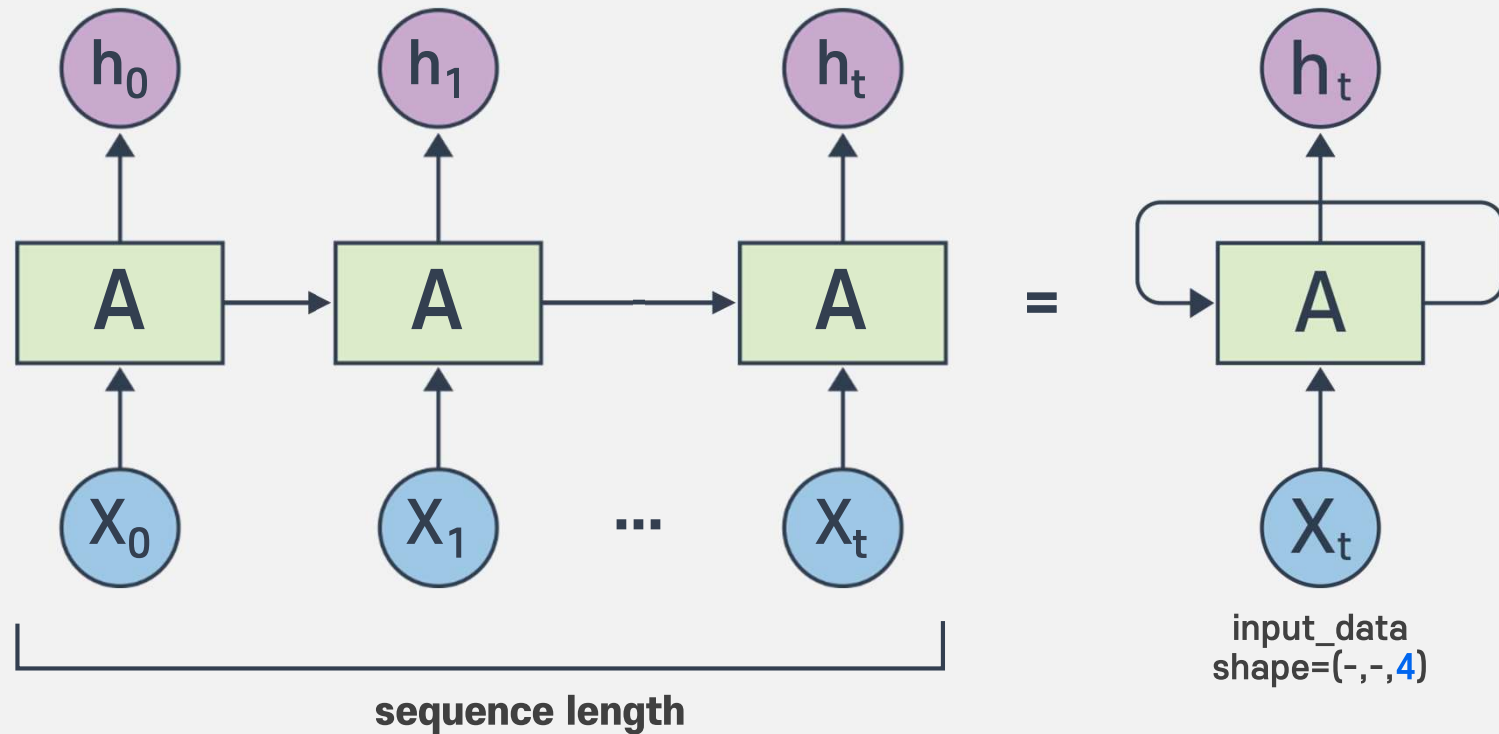
Simple Example (Cont'd)

- Hidden State (cont'd)



Simple Example (Cont'd)

- Sequence Length



Simple Example (Cont'd)

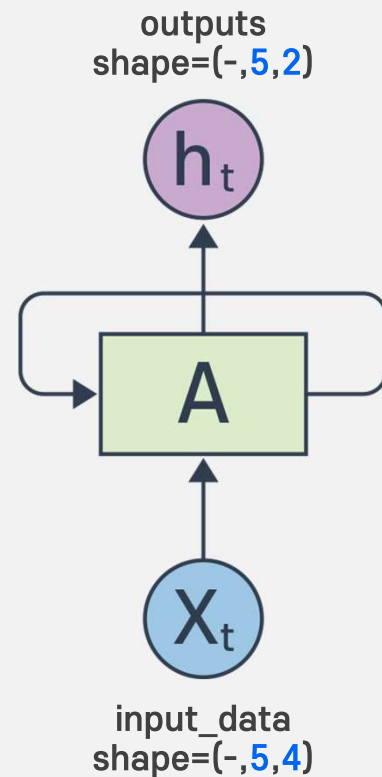
- Sequence Length (cont'd)

h,e,l,l,o

$$\begin{aligned}x_0 &= [1, 0, 0, 0] \\x_1 &= [0, 1, 0, 0] \\x_2 &= [0, 0, 1, 0] \\x_3 &= [0, 0, 1, 0] \\x_4 &= [0, 0, 0, 1]\end{aligned}$$



Automatically Calculated



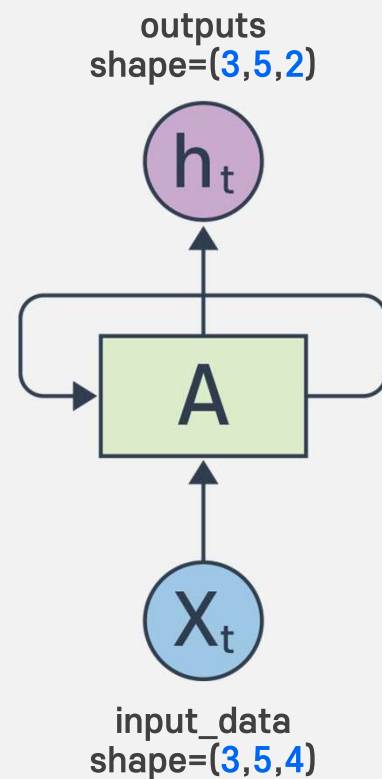
Simple Example (Cont'd)

- Batch Size

$$\begin{bmatrix} h, e, l, l, o \\ e, o, l, l, l \\ l, l, e, e, l \end{bmatrix}$$

Set
batch input

Automatically Calculated





Simple Example (Cont'd)

```
import torch
import numpy as np

# Random seed to make results deterministic and reproducible
torch.manual_seed(0)

# declare dimension
input_size = 4
hidden_size = 2

# sequential example
# shape : (3, 5, 4)
h = [1, 0, 0, 0]
e = [0, 1, 0, 0]
l = [0, 0, 1, 0]
o = [0, 0, 0, 1]
input_data_np = np.array([[h, e, l, l, o], [e, o, l, l, l], [l, l, e, e, l]], dtype=np.float32)
```



Simple Example (Cont'd)

```
# transform as torch tensor
input_data = torch.Tensor(input_data_np)

# declare RNN
rnn = torch.nn.RNN(input_size, hidden_size)

# check output
outputs, _status = rnn(input_data)
print(outputs)
print(outputs.size())
```