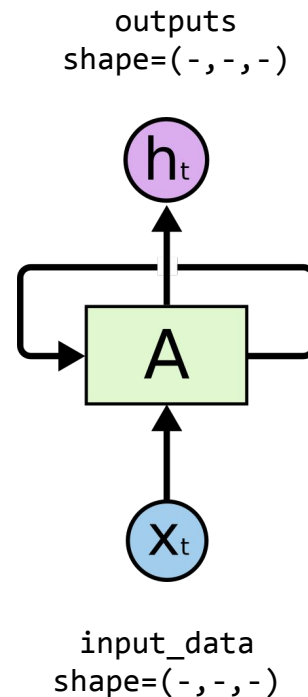


# RNN - Basics

- 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)
```

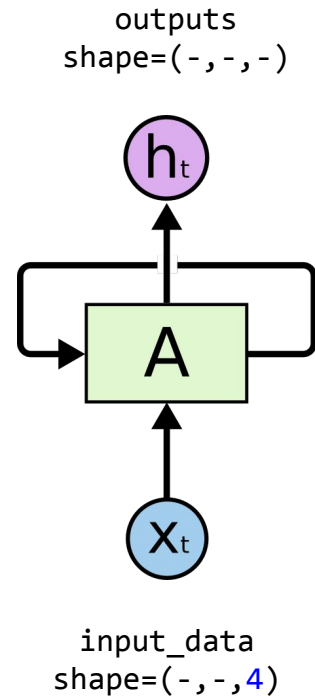


# Example : Input

“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

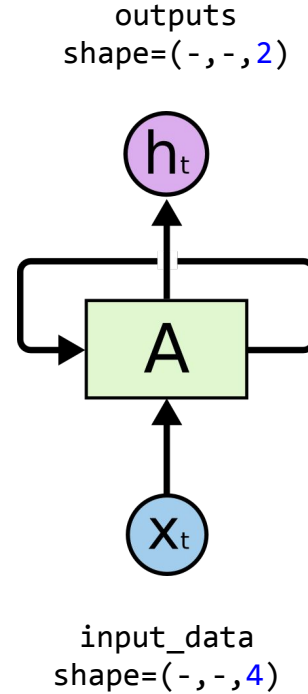


# Example : Hidden State

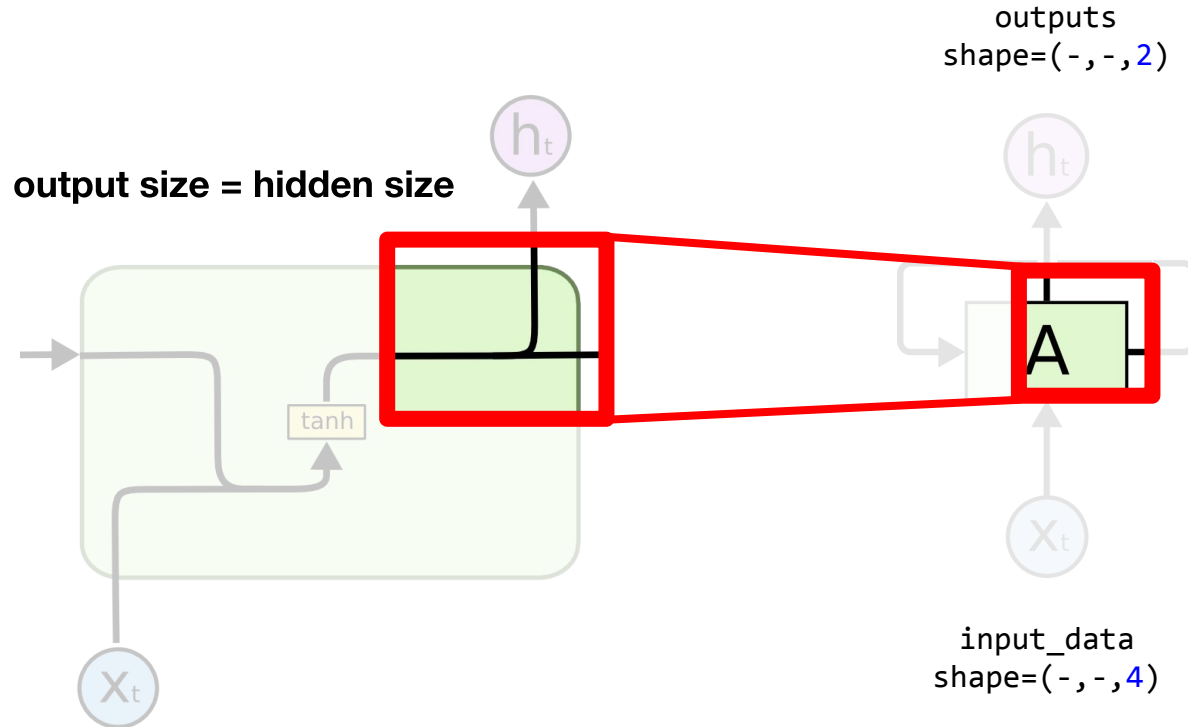
desirable output size



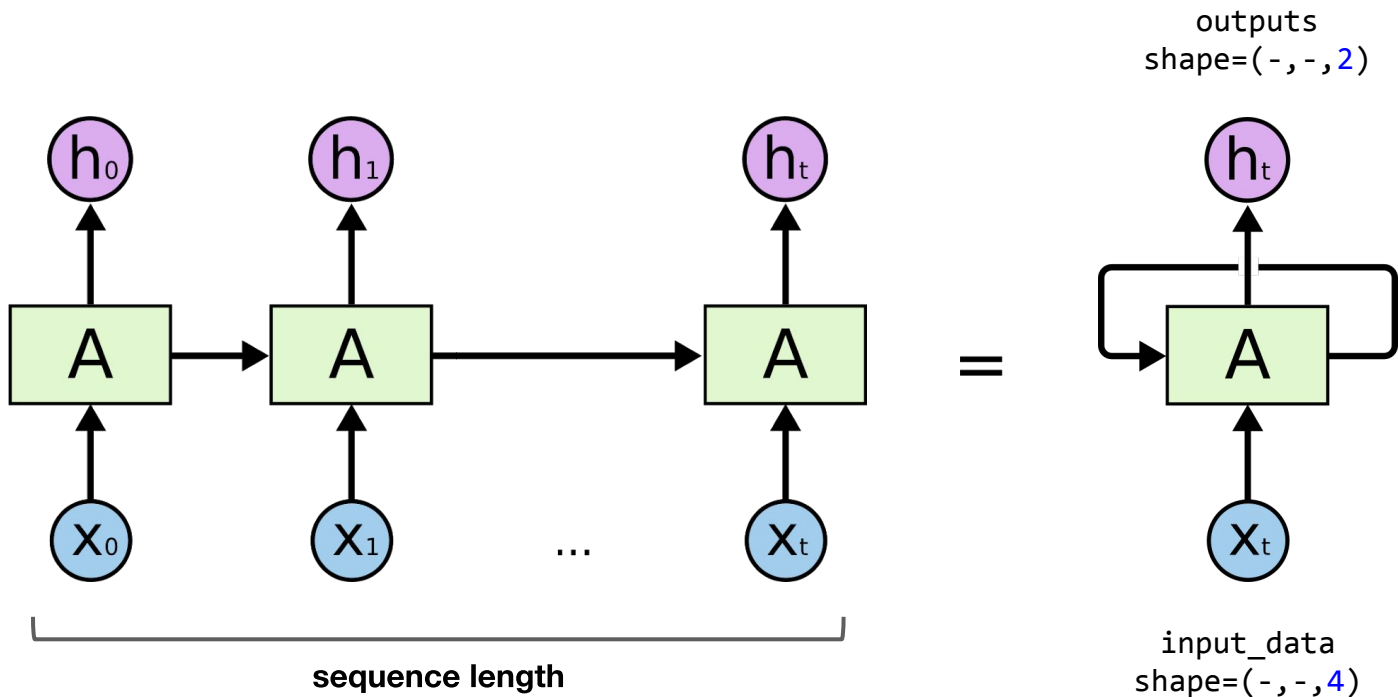
hidden\_size = 2



# Example : Hidden State



# Example : Sequence Length



# Example : Sequence Length

h, e, l, l, o

$$x_0 = [1, 0, 0, 0]$$

$$x_1 = [0, 1, 0, 0]$$

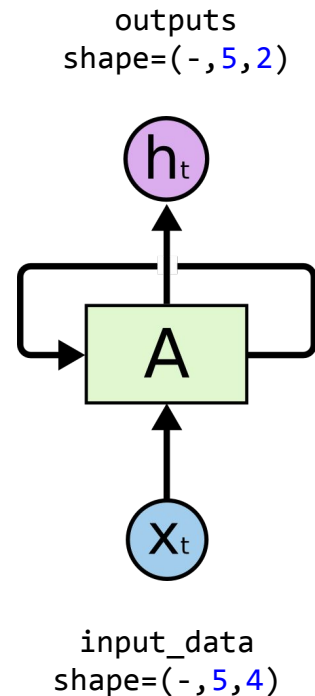
$$x_2 = [0, 0, 1, 0]$$

$$x_3 = [0, 0, 1, 0]$$

$$x_4 = [0, 0, 0, 1]$$



**Automatically Calculated**

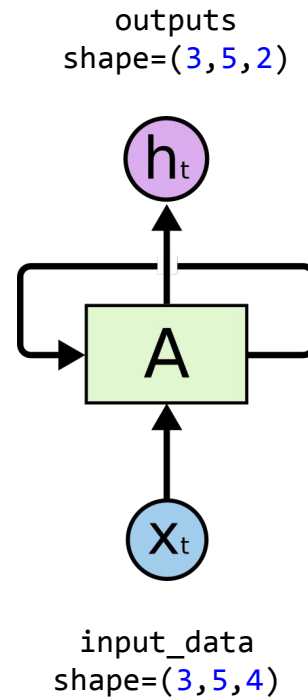


# Example : Batch Size

$$\begin{pmatrix} \text{h, e, l, l, o} \\ \text{e, o, l, l, l} \\ \text{l, l, e, e, l} \end{pmatrix}$$

Set  
batch input

Automatically Calculated





# Example : Run

```
import torch
import numpy as np

input_size = 4
hidden_size = 2

# 1-hot encoding
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)

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

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

outputs, _status = rnn(input_data)
```

# What's Next?

- Train RNN in PyTorch