自然語言處理與應用
Natural Language Processing and Applications

Recurrent Neural Networks

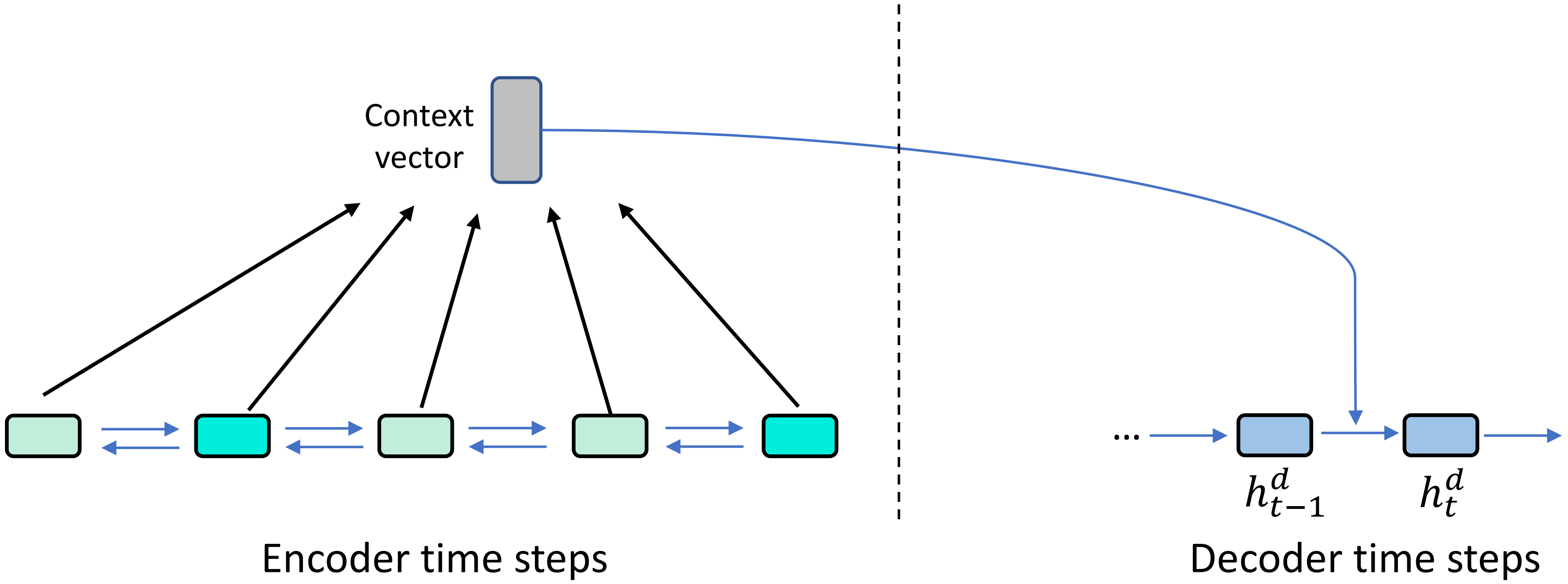
Instructor: 林英嘉 (Ying-Jia Lin)
2025/03/10

Course GitHub

Slido # NLP_RNN

自然語言處理基本功 (序列模型)

- Week 4 – Week 5
  - 遞迴神經網路 / PyTorch Tutorial
  - 注意力機制以及 Sub-word 分詞法



Outline

- Deep learning fundamentals
- RNN (Recurrent Neural Network)
- LSTM (Long Short-Term Memory)

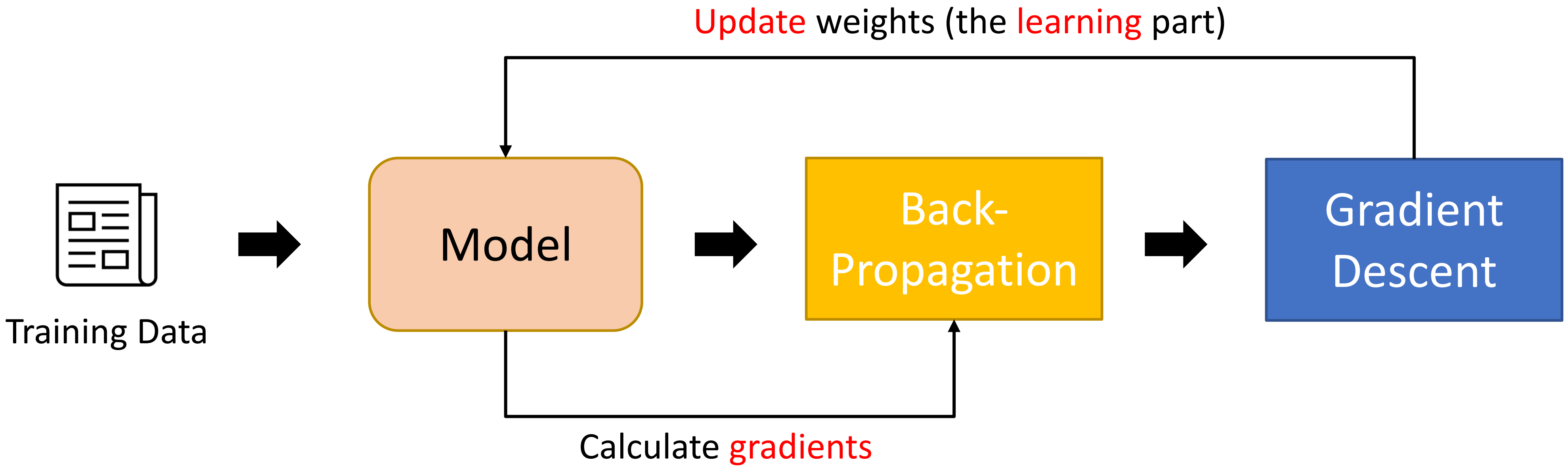
Basic Definitions

- In a deep learning project, there are some fundamental components.

1. Model
   - A model refers to the architecture or structure used to represent relationships between input data and output predictions.
2. Optimizer
   - An optimizer is an algorithm used to adjust the parameters of the model during training in order to minimize the error between predicted and actual output values.
3. Loss function
   - A loss function (objective function) measures the difference between the predicted output of a model and the true target output.

Training Process of a Deep Learning Model

- 深度學習模型被訓練的流程



Training Neural Networks

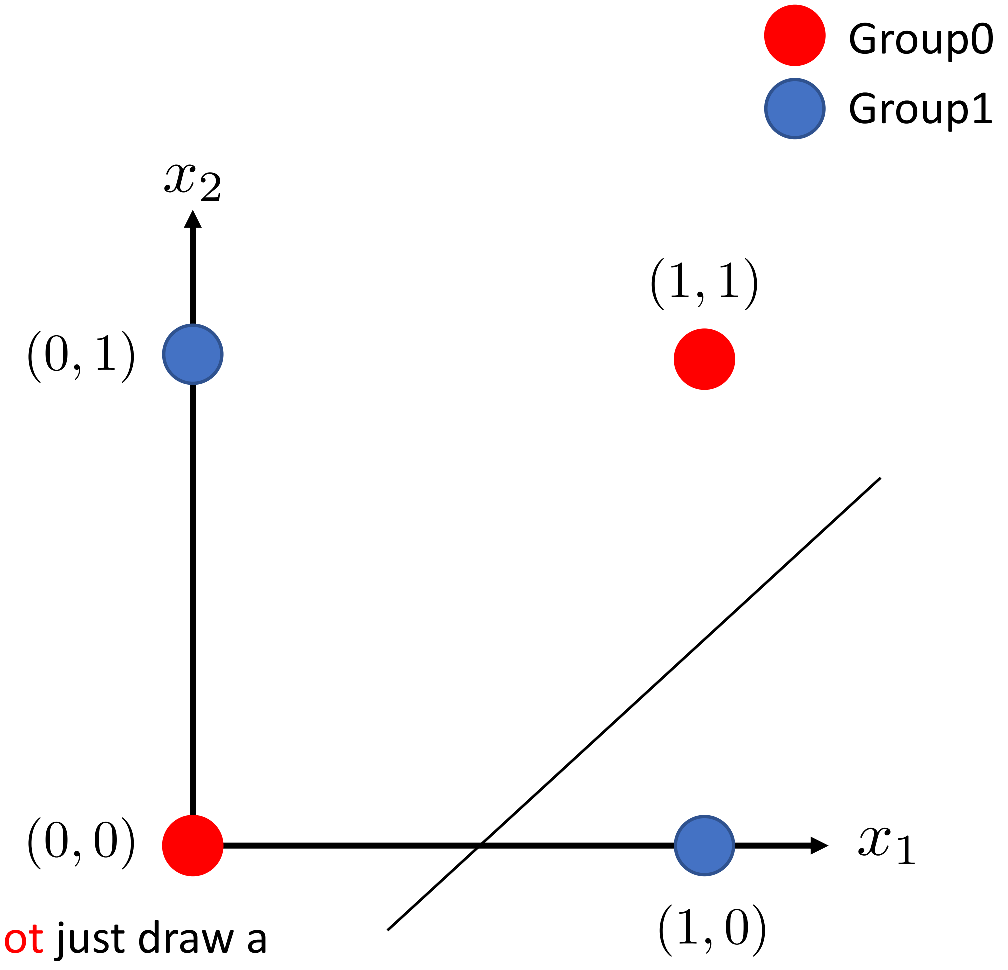
The training process typically involves the following steps:

1. Data Preparation: Prepare training and testing datasets.
2. Model Construction: Construct the model using a deep learning framework (TensorFlow, PyTorch, ...)
3. Loss Function Definition: Select an appropriate loss function. (Cross-entropy, Logloss, ...)
4. Optimizer Selection: Choose a suitable optimization algorithm. (Adam, SGD, ...)
5. Model Training: Train the model using the training dataset.
6. Model Evaluation: Evaluate the trained model using the testing dataset. (F1, LCS, ...)

XOR Problem

XOR: Exclusive-or

x1	x2	XOR
0	0	0
0	1	1
1	0	1
1	1	0



We cannot just draw a line to classify two groups!

https://www.youtube.com/watch?v=S9Kjwp6AUHI
https://www.youtube.com/watch?v=GPUnb61iLGc

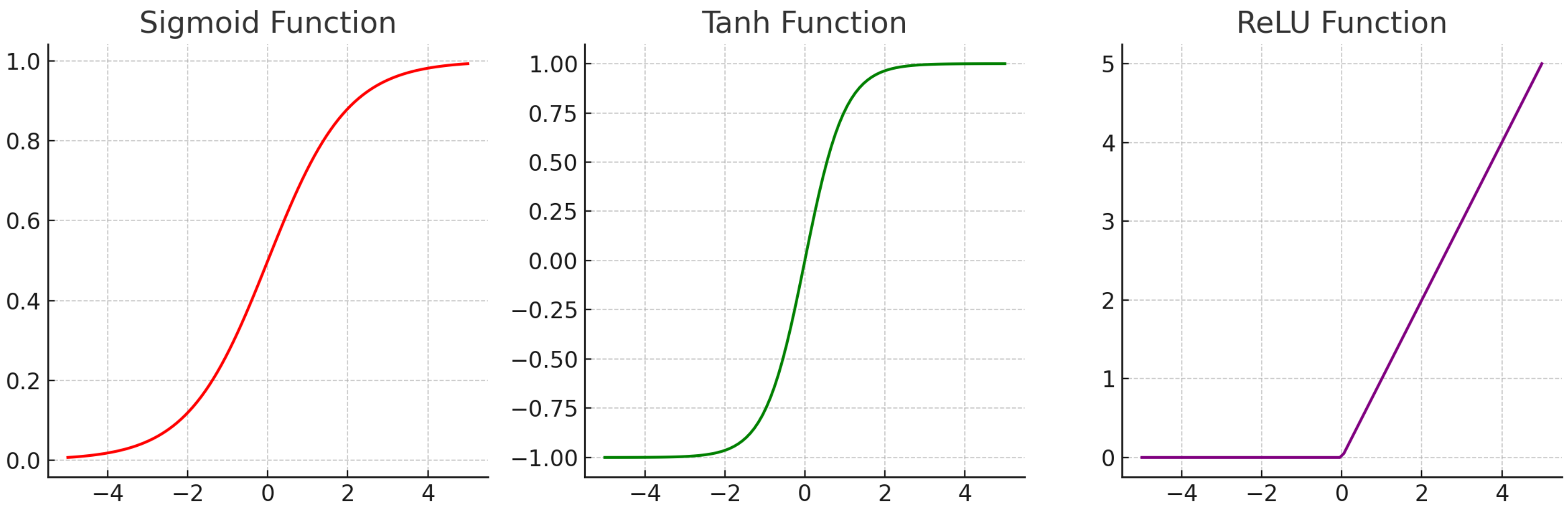
Activation functions

- The core idea of using activation functions is to introduce nonlinearity into neural networks.
- Neural network models aim to avoid the final processing stage being merely a linear transformation of the inputs, so that the model is able to have good performance on complex problems.
- Commonly used activation functions including Sigmoid, ReLU, tanh, GeLU...

Activation functions

	Range	Applications
softmax	[0, 1] (sum up to 1)	It is commonly used in multi-class classification tasks where the model needs to predict the probability distribution over classes.
sigmoid	[0, 1]	It often used in binary classification tasks where a threshold is needed to predict probabilities of belonging to one of the two classes.
tanh	[-1, 1]	It is often preferred over sigmoid for hidden layers as it produces zero-centered output
ReLU	[0, +inf]	It introduce nonlinearity and avoid gradient vanishing

Activation functions



Softmax: https://datascience.stackexchange.com/questions/57005/why-there-is-no-exact-picture-of-softmax-activation-function

Softmax

- 採用 exponential -> 大的數值更大，小的數值更小
  - 有助於梯度下降

$$y = \frac{\mathrm{x}}{\sum_{\mathrm{j}} \mathrm{x_j}} = [0.5,0.25,0.25]$$

$$\mathrm{Softmax}(x_i) = \frac{e_i^x}{\sum_j e^{x_j}} = [0.665,0.244,0.090]$$

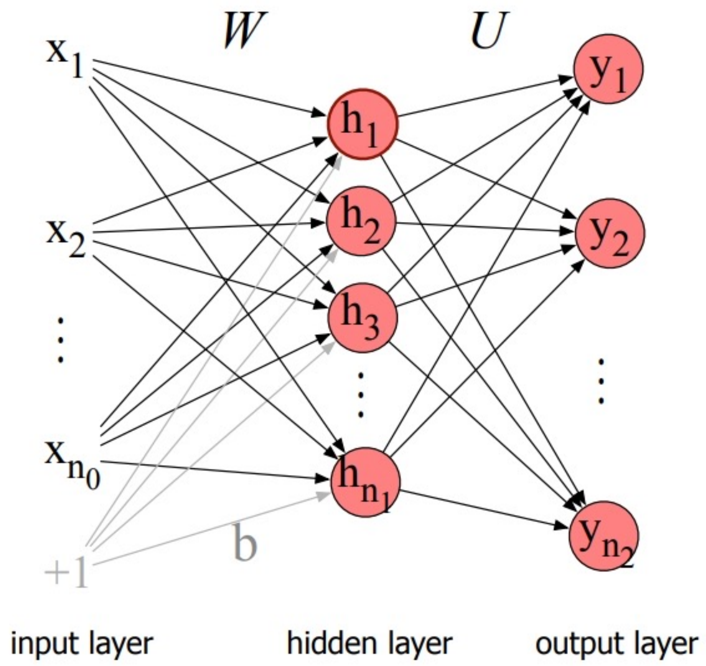
Feedforward network

- Feedforward network (FFN) is a multilayer feedforward network in which the units are connected with no cycles.

$$h = \sigma(Wx + b)$$

$$z = Uh$$

$$y = softmax(z)$$

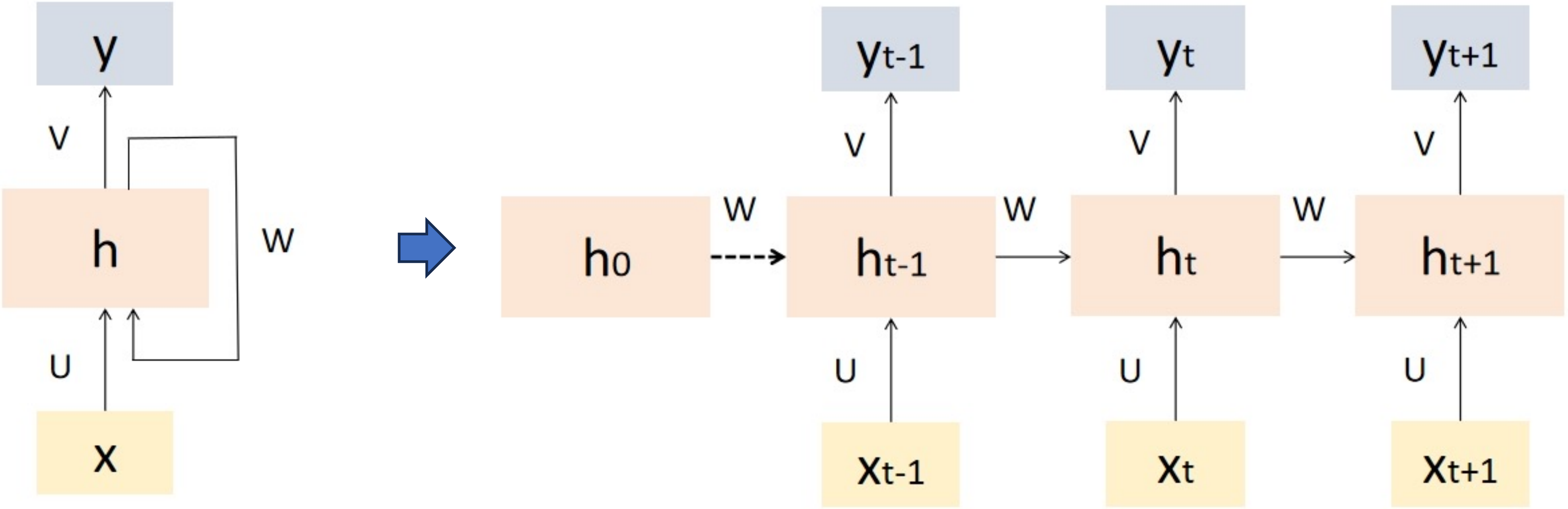


Short comings of FFN

- **Lack of Sequence Modeling:**
  - In NLP tasks, understanding the sequence of words and their dependencies is crucial for accurate predictions.
- **Fixed Input Size:**
  - FFNs require fixed-size inputs, which can be problematic for NLP tasks where input sequences vary in length.
- **Limited Contextual Information:**
  - Many NLP tasks benefit from capturing long-range dependencies and understanding the broader context of a text, which is better addressed by models capable of modeling sequential data effectively.

RNN

- Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to handle sequential data by capturing temporal dependencies.
  - The same set of weights and biases are used across all time steps



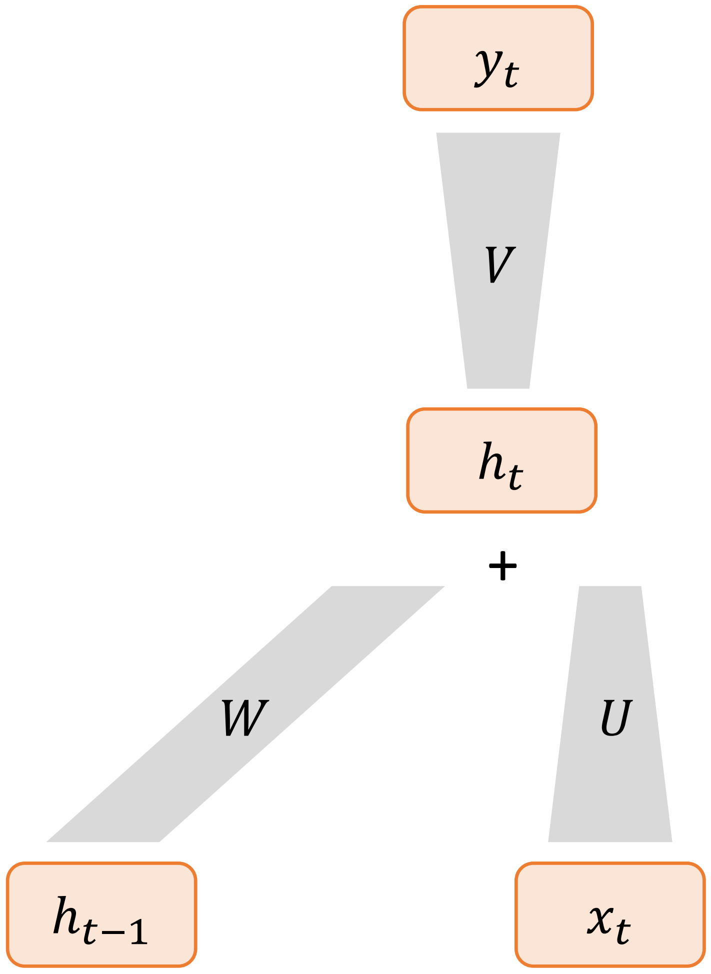
RNN

- The equation on step t is:

$$y_t = g(Vh_t)$$

$$h_t = f(Ux_t + Wh_{t-1})$$

, where f and g are activation functions.



Properties of RNNs

- **Sequential Processing:**
  - RNNs handle sequences, allowing them to model temporal dependencies in data.
- **Recurrent Connections:**
  - RNNs maintain internal memory, facilitating the capture of long-term dependencies.
- **Parameter Sharing:**
  - RNNs share parameters across time steps, enhancing efficiency in learning sequential data.
- **Vanishing Gradient Problem:**
  - Traditional RNNs may face vanishing gradient issues, hindering learning of long-term dependencies.

Name Entity Recognition

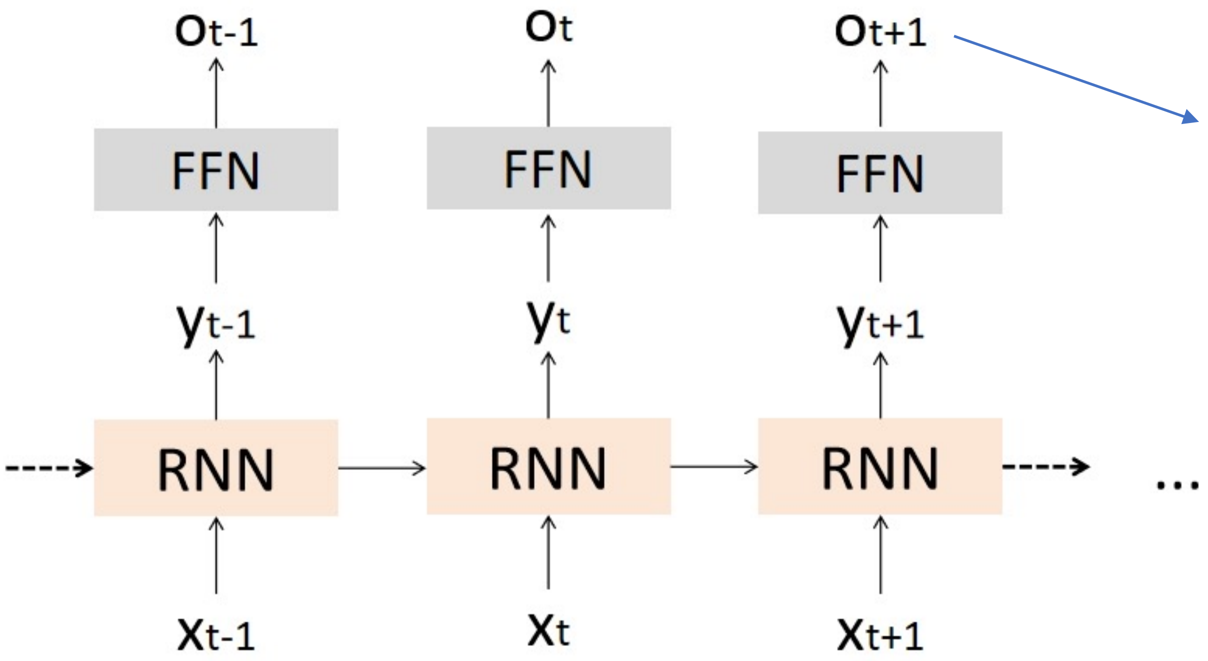
- Name Entity Recognition (NER) is a fundamental task in NLP.
- The model needs to identify named entities within the sequence, such as countries, organizations, and individuals.

	My flight to	New York		is at	5pm
		LOC			TIME
Strategy: BIO		B-LOC	I-LOC		B-TIME
Strategy: BIOES		B-LOC	E-LOC		S-TIME

	Meaning
B	Beginning
I	Inside
O	Outside
E	End
S	Single

RNNs for NER

- In token classification task, every output should be mapped to a one-hot vector.
- A feed forward network is added to the RNN model.



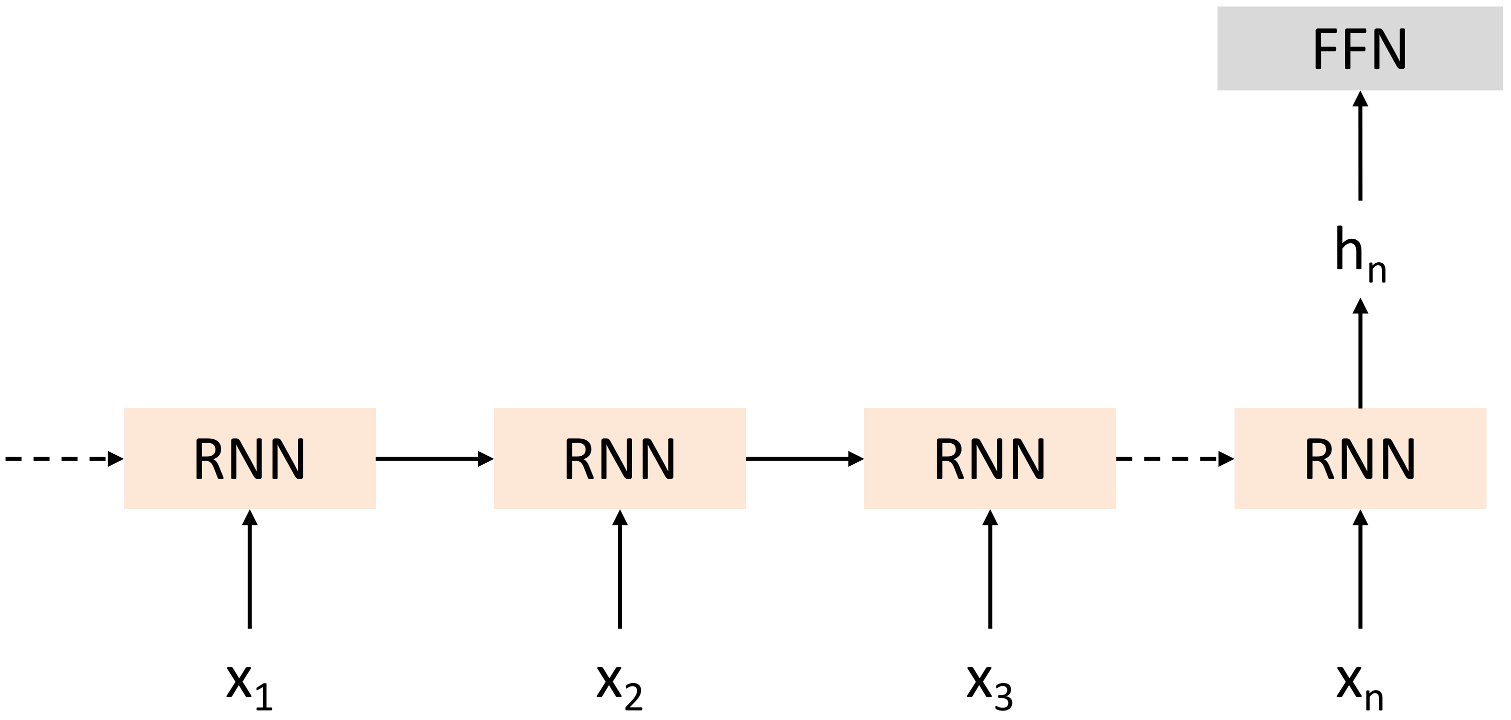
(Nx5) matrix

	Meaning
B	Beginning
I	Inside
O	Outside
E	End
S	Single

(取最大值)

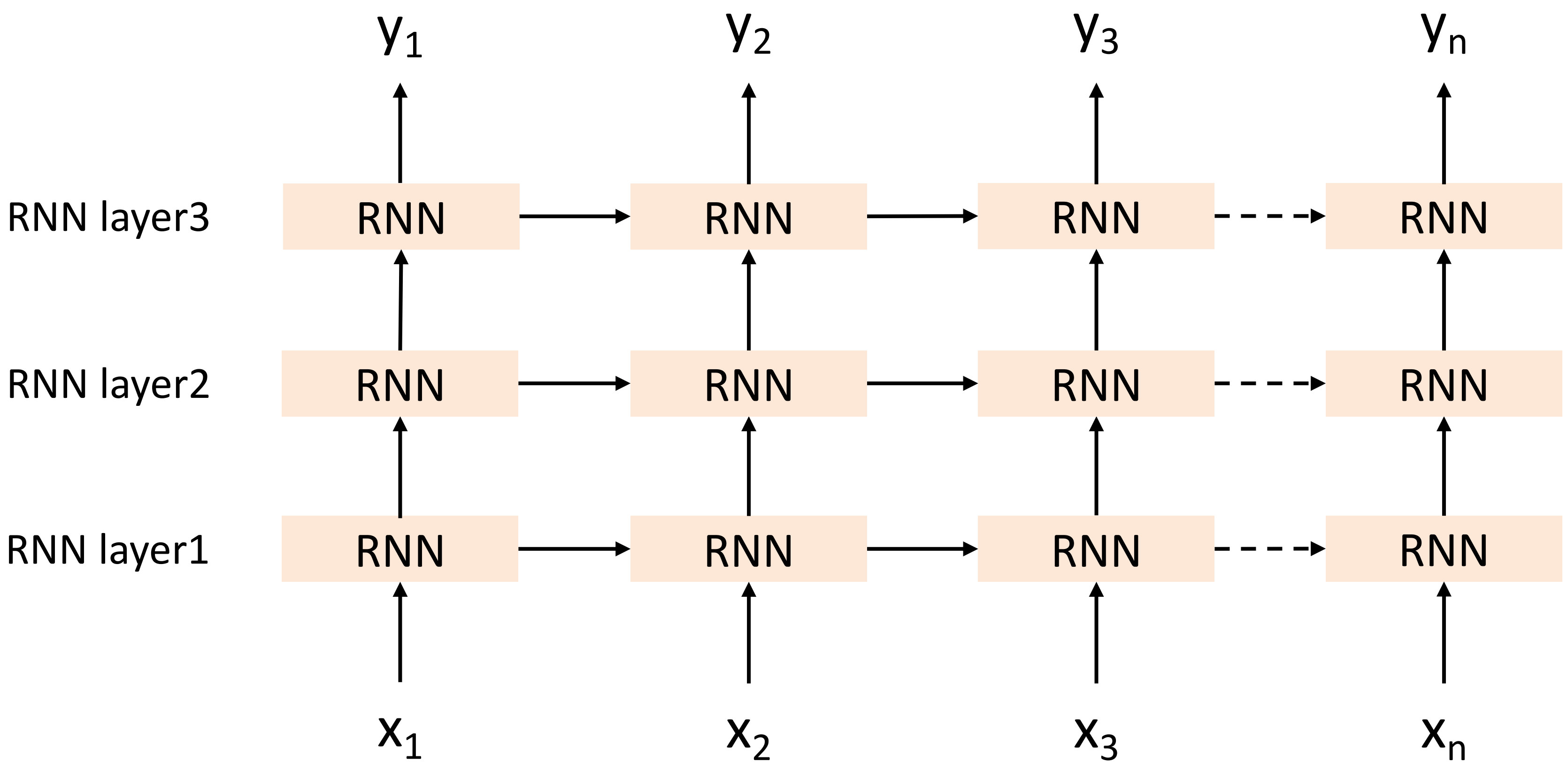
RNNs for Sequence Classification

- RNNs classify the entire sequences rather than the tokens within them.
- Take the hidden layer for the last token of the text.



Stacked RNNs

- Stacked RNNs consist of multiple networks where the output of one layer serves as the input to a subsequent layer

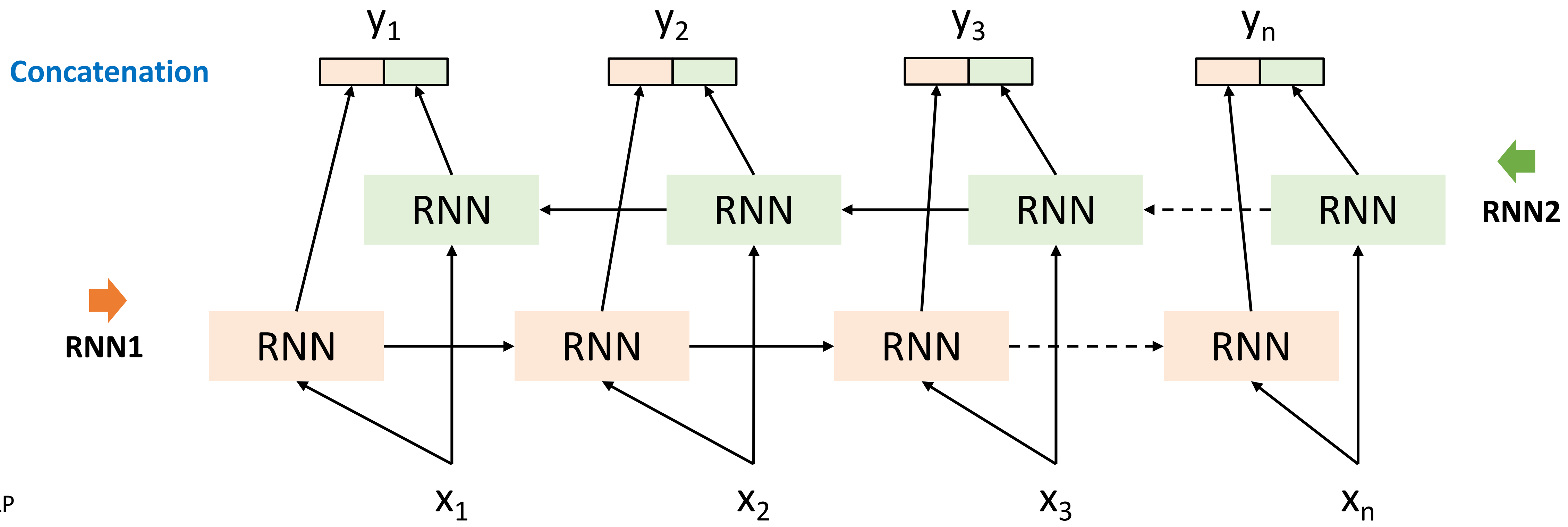


Stacked RNNs

- Stacked RNNs generally outperform single-layer networks.
  - The network induces representations at differing levels of abstraction across layers
  - The initial layers of stacked networks induce representations that serve as useful abstractions for further layers
- However, as the number of stacks is increased the training costs rise quickly.

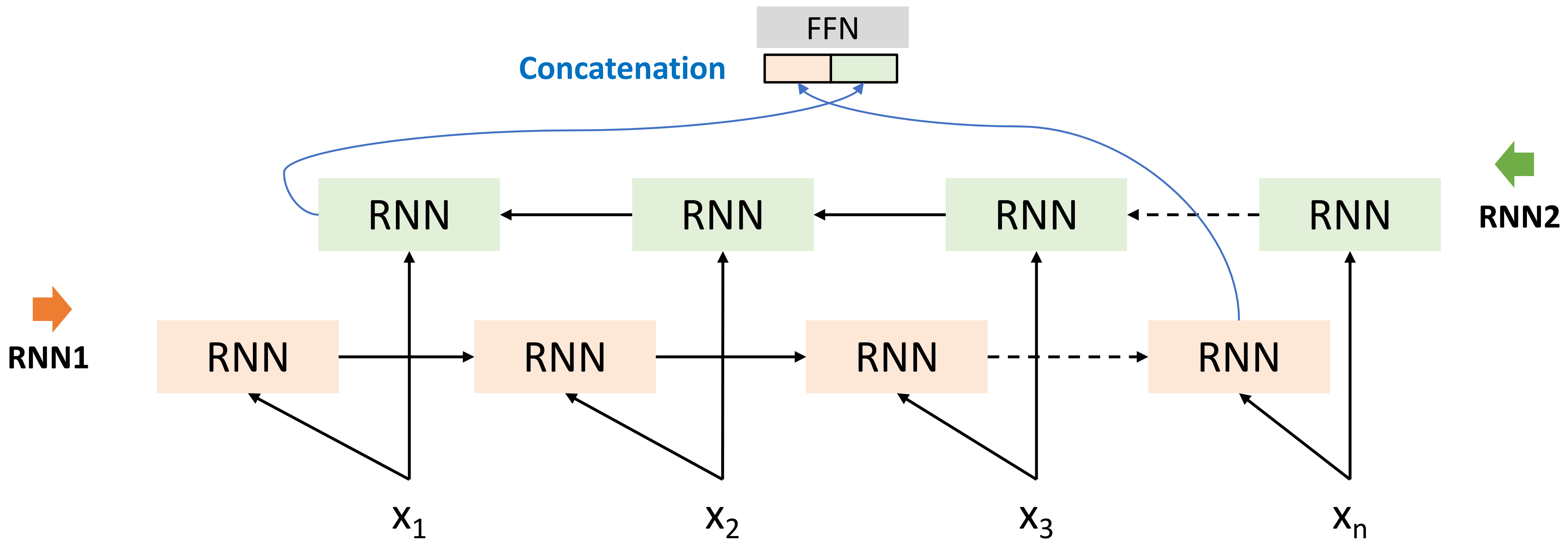
Bidirectional RNNs

- In many applications, RNNs have to access to the entire input sequence.
- Bidirectional RNN was introduced.
  - It combines two independent bidirectional RNNs, one where the input is processed from the start to the end, and the other from the end to the start



Bidirectional RNNs for Sequence Classification

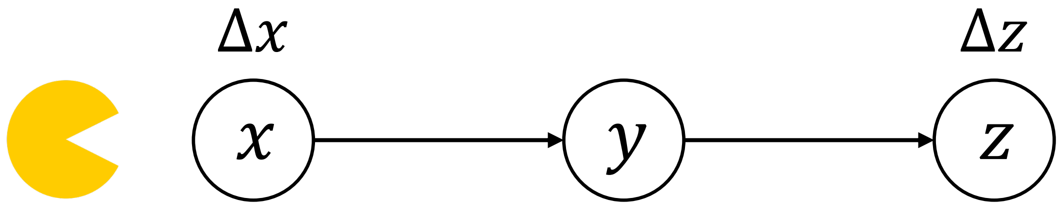
- The final hidden units from the forward and backward passes are combined to represent the entire sequence.
- This combined representation serves as input to the subsequent classifier.



Shortage of standard RNNs

- Gradient vanishing (梯度消失) problem
- Long-term dependencies cannot be handled.
  - RNNs can only remember local relationships.
  - This is result from back-propagation through-time (BPTT) during training.

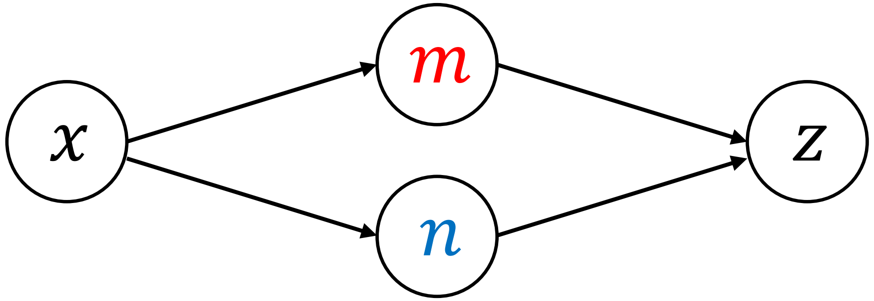
(Calculus) Chain Rule - 1



$$\frac{dz}{dx} = \underline{\frac{dz}{dy}}\,\underline{\frac{dy}{dx}}$$

- $\frac{dz}{dx}$ 代表 x 對 z 的影響，但 <u>x 的變化會影響到 y</u>，接著 <u>y 影響到 z</u>

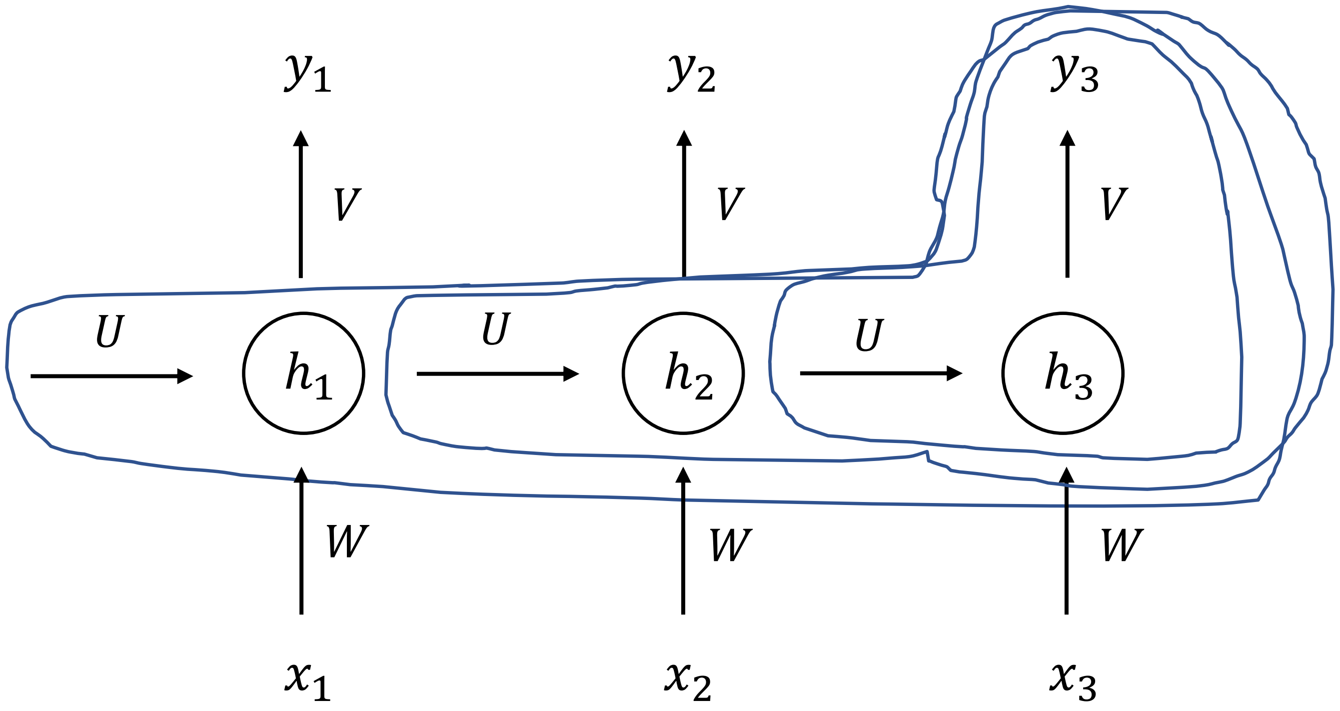
(Calculus) Chain Rule - 2



$$\frac{dz}{dx} = \frac{dz}{dm}\frac{dm}{dx} + \frac{dz}{dn}\frac{dn}{dx}$$

- $\frac{dz}{dx}$ 代表 x 對 z 的影響
  - 但 z 的變化由 m 和 n 共同影響

Back-propagation through-time (BPTT)



$$\frac{\partial \mathcal{L}}{\partial U} = \frac{\partial \mathcal{L}}{\partial y_3} \cdot \frac{\partial y_3}{\partial h_3} \cdot \frac{\partial h_3}{\partial U} + \frac{\partial \mathcal{L}}{\partial y_3} \cdot \frac{\partial y_3}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial U} + \underbrace{\frac{\partial \mathcal{L}}{\partial y_3} \cdot \frac{\partial y_3}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial U}}_{\text{(small)}}$$

Gradient Descent (梯度下降法)

Assume x is a trainable parameter (weight), f is a differentiable function:

Gradient descent: $$x' = x - \eta \nabla_x f(x)$$

η is the learning rate used for gradient descent.

Back-propagation

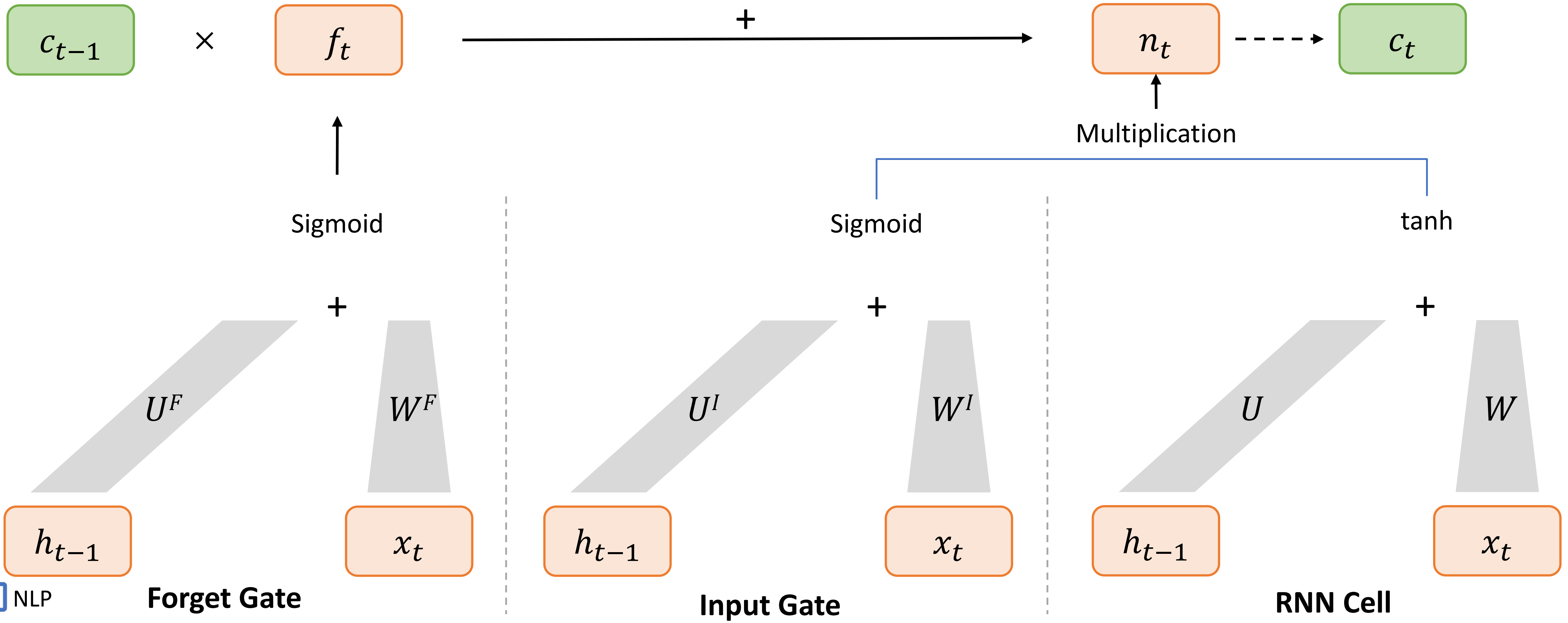
- Videos from the deep learning course
  - https://youtu.be/xtP6g116-Fg?si=edgUymIATpALUYcw
  - https://youtu.be/6xHIgJU4Csg?si=UbSVQOk3bFRScgub

Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

LSTM (Long Short-Term Memory)

- LSTM uses 4 times of weights as a standard RNN.

$$c_t = c_t \times f_t + n_t$$

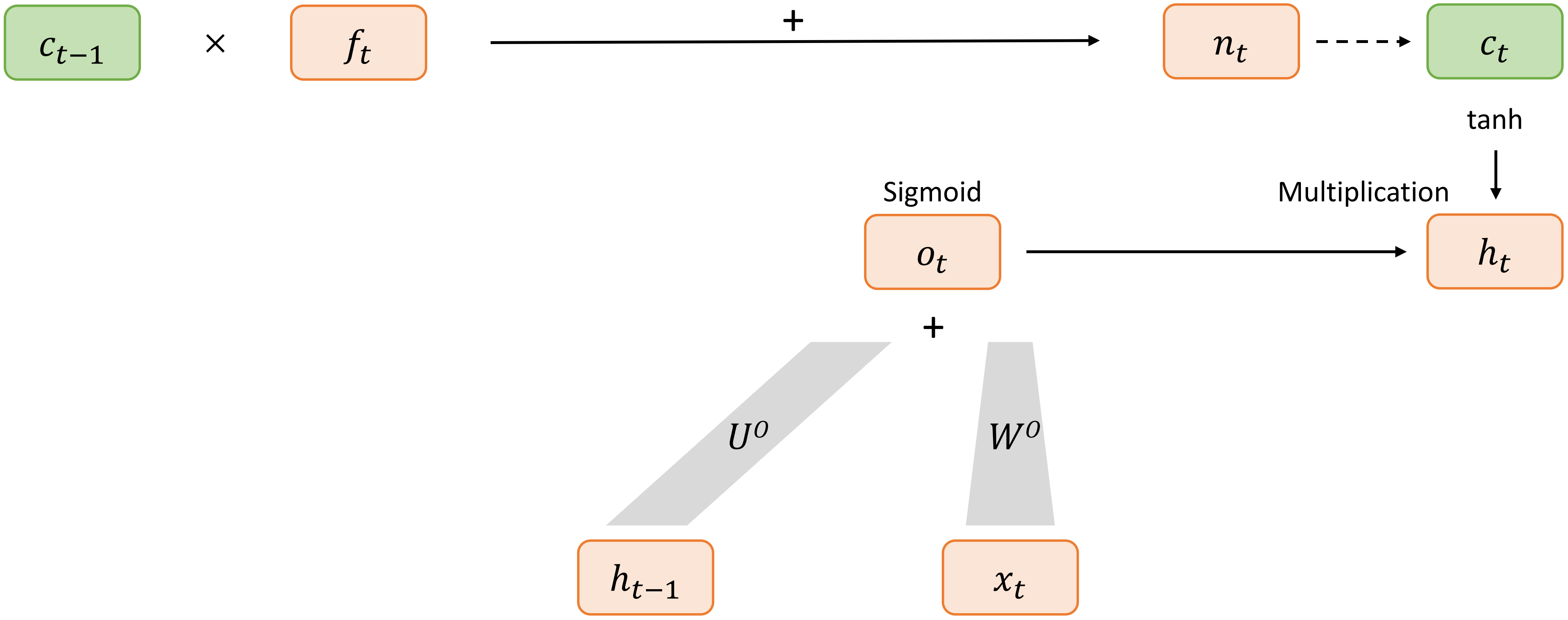


Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

LSTM (Long Short-Term Memory)

➤ LSTM uses 4 times of weights as a standard RNN.

$$c_t = c_t \times f_t + n_t$$



Output Gate

RNN vs. LSTM

	Memory Mechanism	Handles Long Dependencies?	Gradient Vanishing?
RNN	Hidden state h_t	Worse	Yes
LSTM	Memory state c_t + Hidden state h_t	Better	Much less than a standard RNN

AD

【人工智慧學系辦公室轉發】

主旨：開放報名!!【波蘭波茲南理工大學出國交換說明會】

智慧運算學院與人工智慧研究中心共同辦理「波蘭波茲南理工大學出國交換說明會」，誠摯邀請有興趣的同學踴躍參加。

此次說明會很榮幸邀請到兩位波蘭波茲南理工大學主管擔任講者：

Ÿ Prof. Mariusz Głąbowski, Ph.D., D.Sc., Vice-Rector for International Relations(國際事務副校長)

Ÿ Associate Prof. Anna Kobusinska, Ph.D., D.Sc., Head of Division of Computing Systems(計算系統部門主管)

兩位講者將在說明會中分享以下內容：

✅波茲南理工大學的第一手資訊

✅波蘭留學與獎學金機會

✅交換計畫申請流程與準備要點

活動資訊：

(一)日期：114年3月21日(五) 11:50 - 13:00

(二)地點：長庚大學管理大樓11F人工智慧研究中心

(三)報名網址：https://forms.gle/YcLtXTmddStzx3Ar6

(四) 即日起開放報名，至3月19日截止。名額有限，額滿將提前關閉報名。

(五)說明會提供便當

Thank you!

Instructor: 林英嘉
✉️ yjlin@cgu.edu.tw
TA: 吳宣毅
✉️ m1161007@cgu.edu.tw