18AIC301J: DEEP LEARNING TECHNIQUES

B. Tech in ARTIFICIAL INTELLIGENCE, 5th semester

Faculty: **Dr. Athira Nambiar**
Section: A, slot:D
Venue: TP 804
Academic Year: 2022-22

UNIT-4

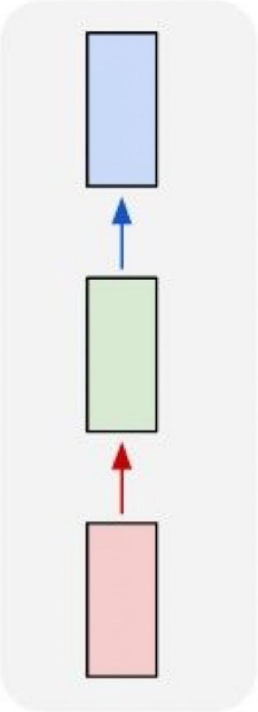
DenseNet Architecture, Transfer Learning
Need for Transfer Learning, Deep Transfer Learning, Types of Deep Transfer learning, Applications of Transfer learning
Transfer learning implementation using VGG16 model to classify images
Sequence Learning Problems, Recurrent Neural Networks
Backpropagation through time, Unfolded RNN, The problem of exploding and vanishing Gradients, Seq to Seq Models
Building a RNN to perform Character level language modeling.
How gates help to solve the problem of vanishing gradients, Long-Short Term Memory architectures
Dealing with exploding gradients, Gated Recurrent Units, Introduction to Encoder Decoder Models, Applications of Encoder Decoder Models
Build a LSTM network for Named Entity recognition.

UNIT-4

DenseNet Architecture, Transfer Learning
Need for Transfer Learning, Deep Transfer Learning, Types of Deep Transfer learning, Applications of Transfer learning
Transfer learning implementation using VGG16 model to classify images
Sequence Learning Problems, Recurrent Neural Networks
Backpropagation through time, Unfolded RNN, The problem of exploding and vanishing Gradients, Seq to Seq Models
Building a RNN to perform Character level language modeling.
How gates help to solve the problem of vanishing gradients, Long-Short Term Memory architectures
Dealing with exploding gradients, Gated Recurrent Units, Introduction to Encoder Decoder Models, Applications of Encoder Decoder Models
Build a LSTM network for Named Entity recognition.

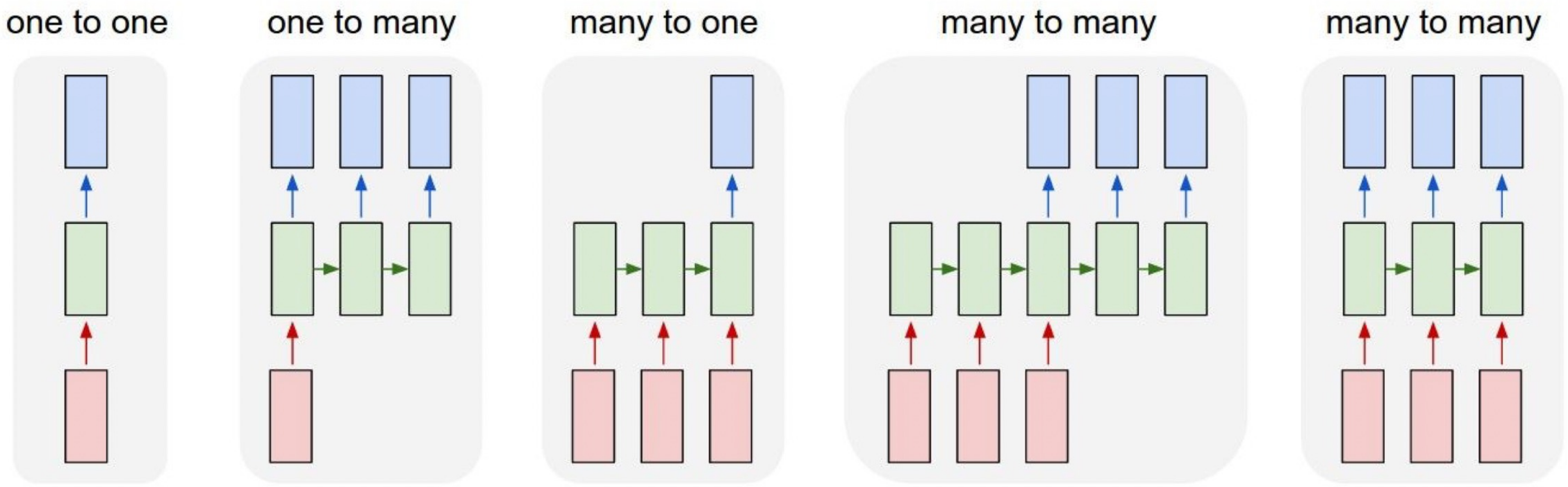
“Vanilla” Neural Network

one to one



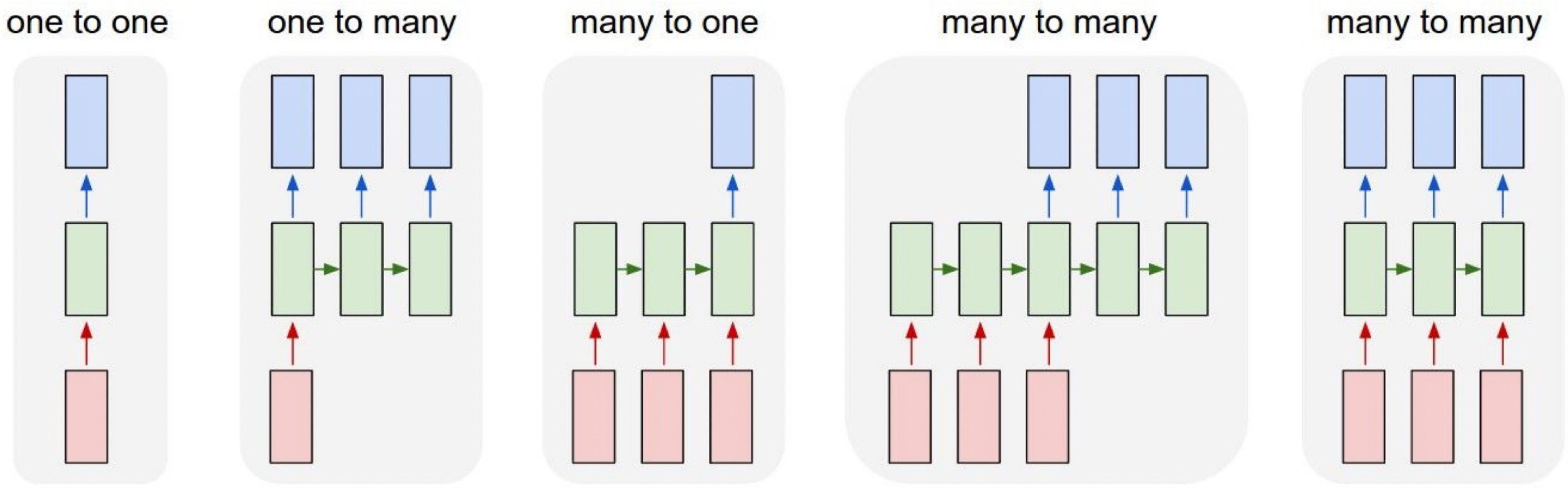
Vanilla Neural Networks

Recurrent Neural Networks: Process Sequences



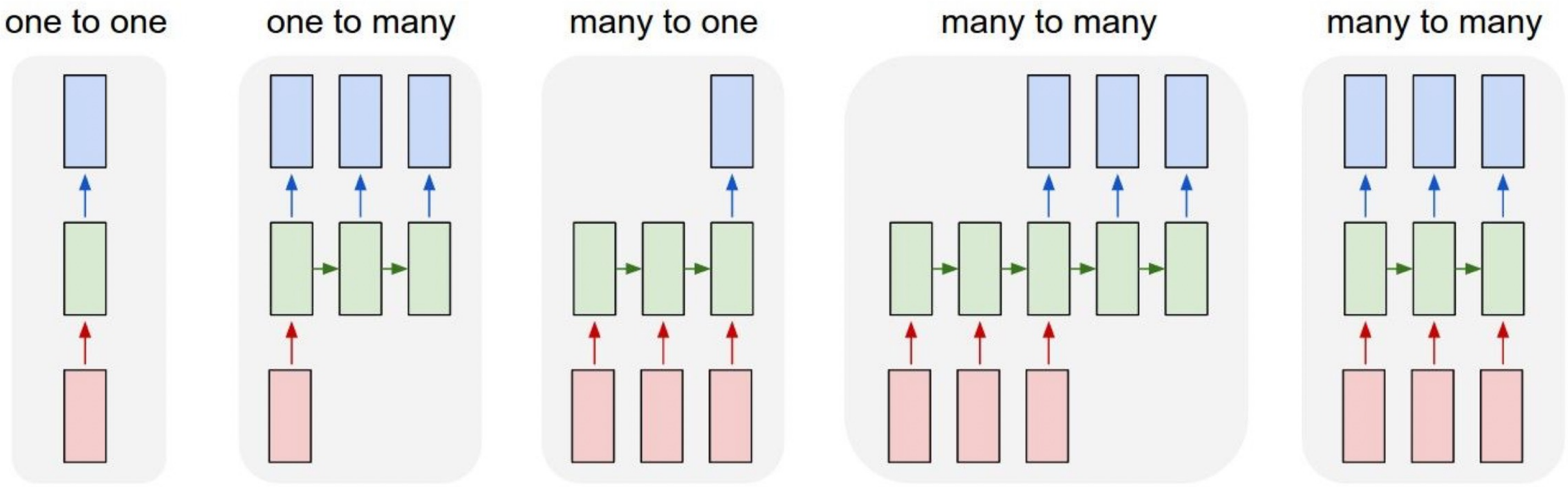
e.g. **Image Captioning**
image -> sequence of words

Recurrent Neural Networks: Process Sequences



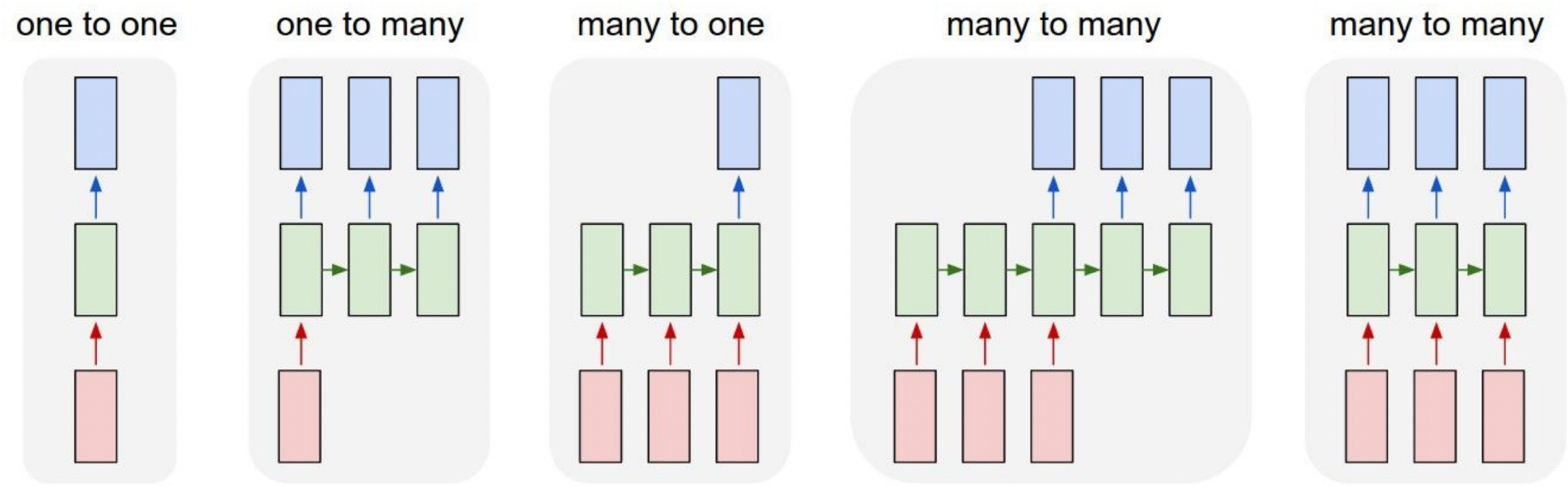
e.g. **Sentiment Classification**
sequence of words -> sentiment

Recurrent Neural Networks: Process Sequences



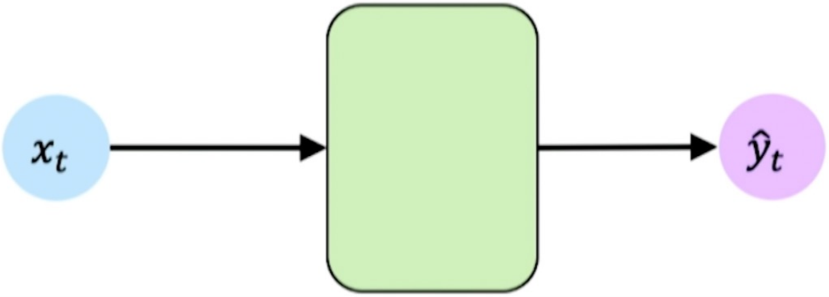
e.g. **Machine Translation**
seq of words -> seq of words

Recurrent Neural Networks: Process Sequences



e.g. **Video classification on frame level**

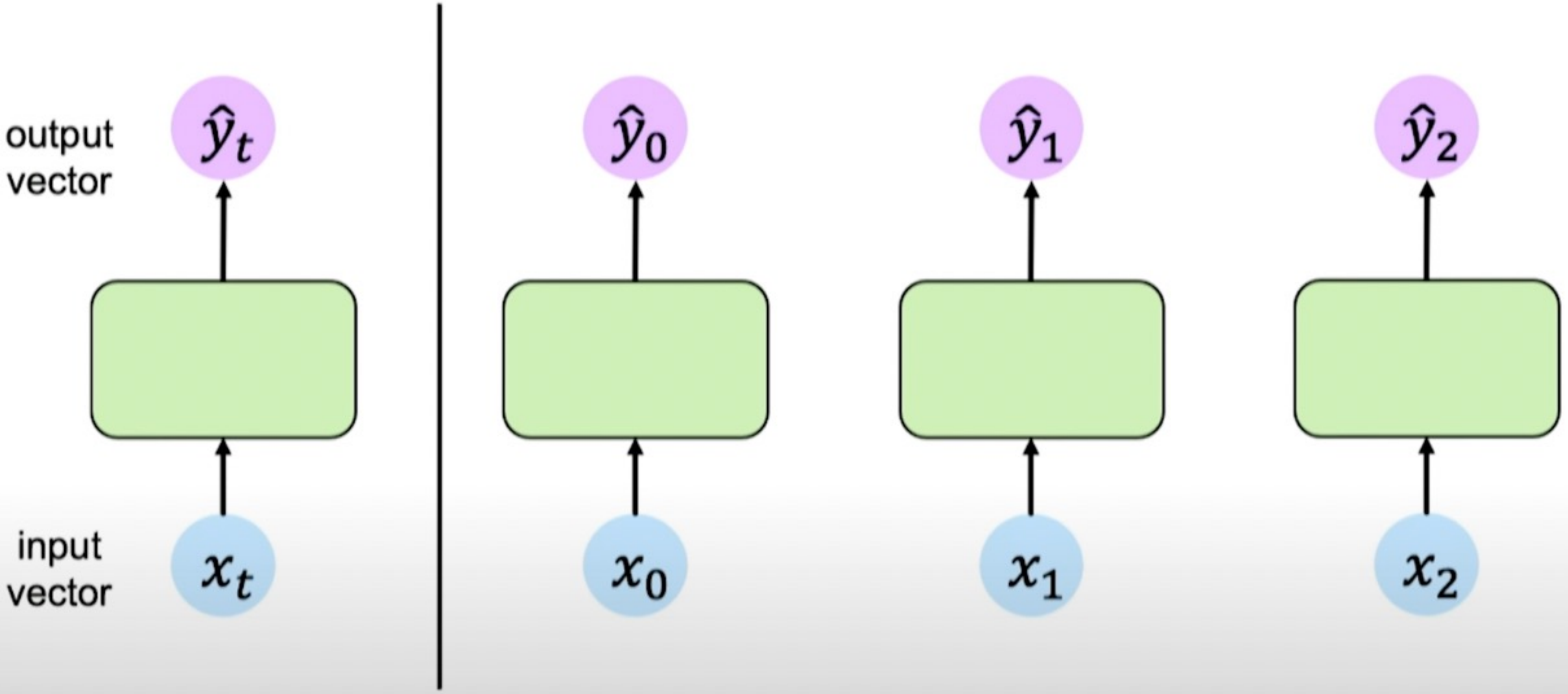
Feed-Forward Networks Revisited

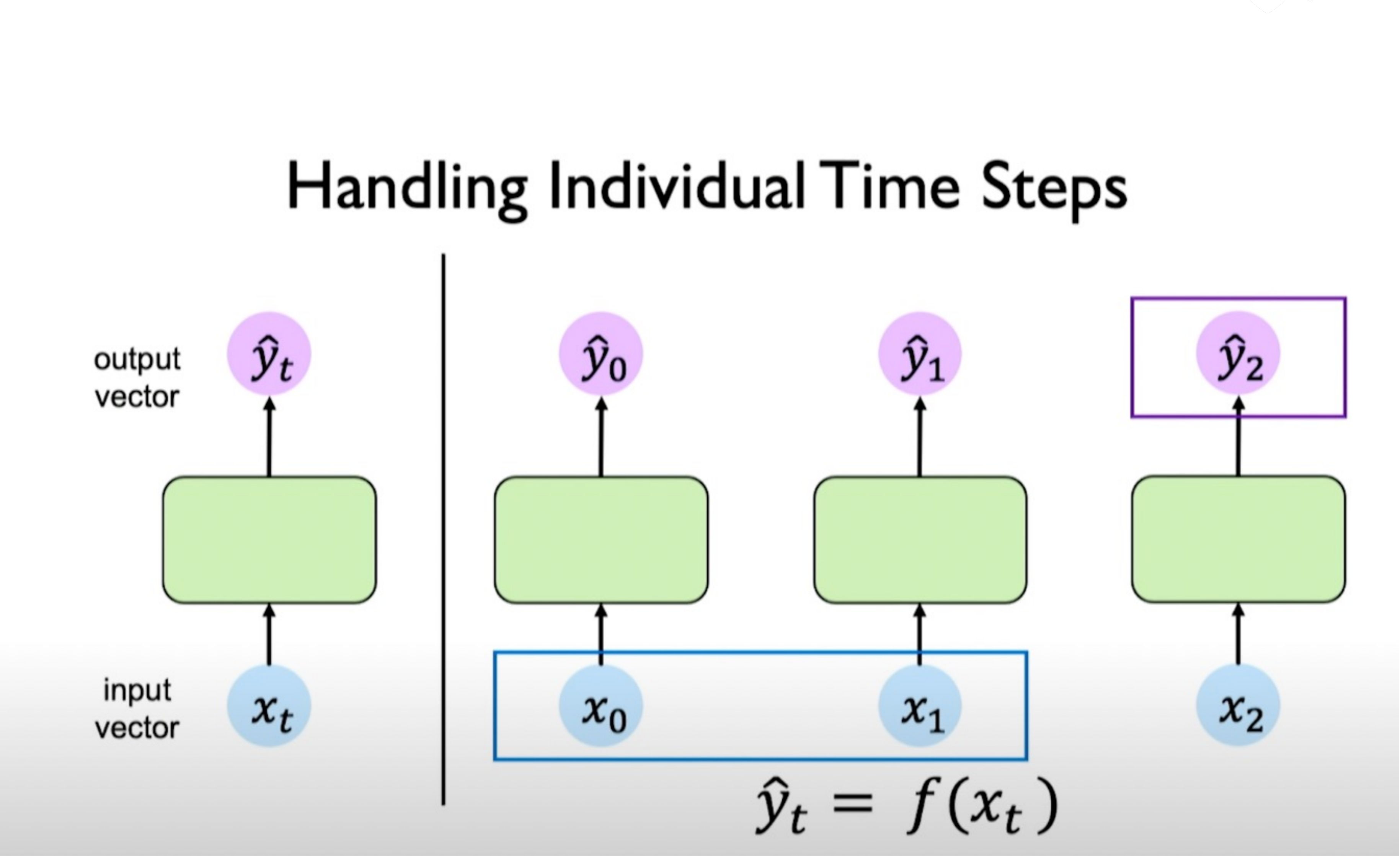


$\boldsymbol{x_t} \in \mathbb{R}^m$

$\hat{\boldsymbol{y}}_t \in \mathbb{R}^n$

Handling Individual Time Steps

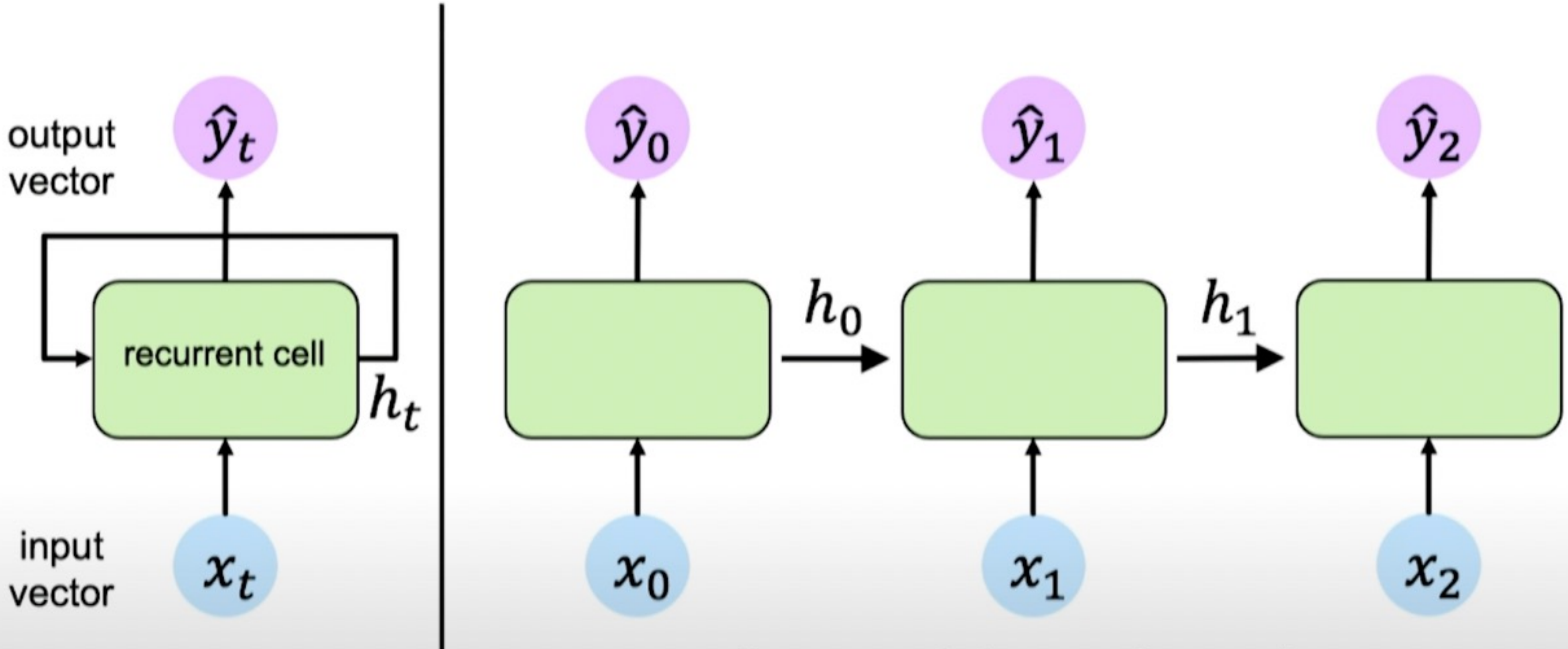




Handling Individual Time Steps

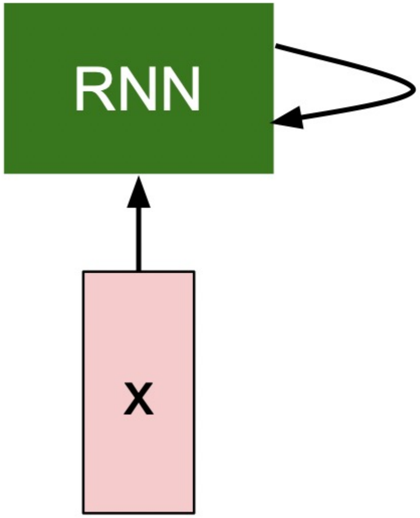
$$\hat{y}_t = f(x_t)$$

Neurons with Recurrence

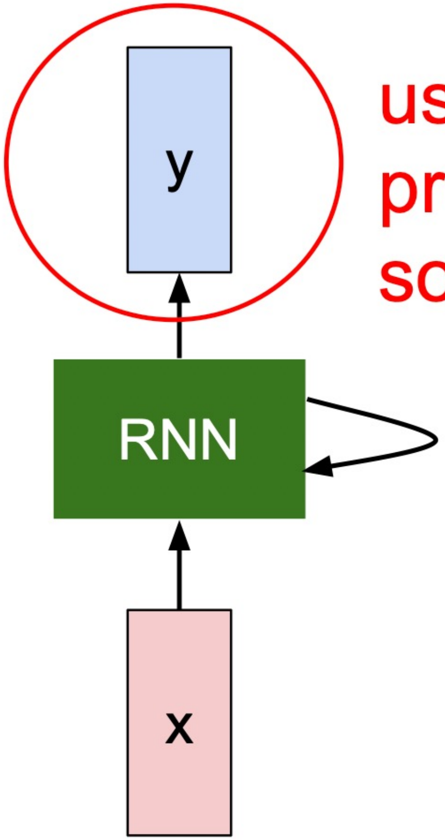


$$\underset{\text{output}}{\underline{\hat{y}_t}} = f(\underset{\text{input}}{\underline{x_t}},\ \underset{\text{past memory}}{\underline{h_{t-1}}})$$

Recurrent Neural Network



Recurrent Neural Network



usually want to predict a vector at some time steps

Recurrent Neural Network

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

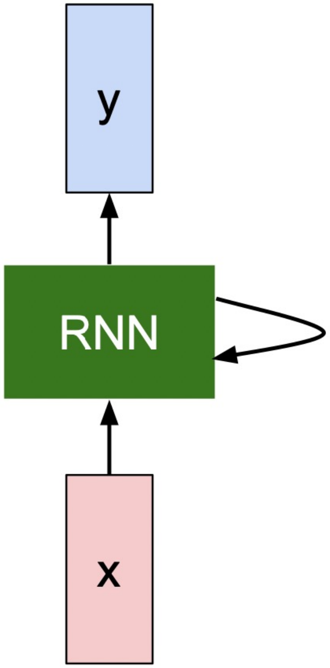
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W

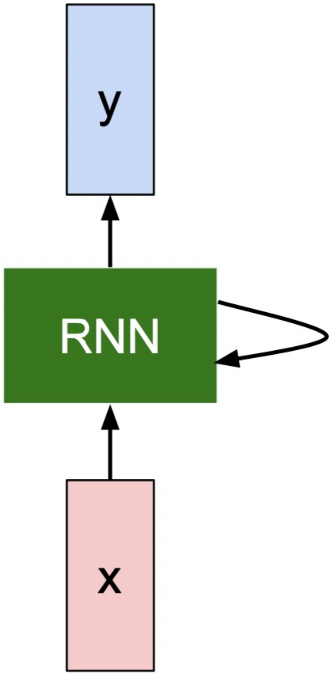


Recurrent Neural Network

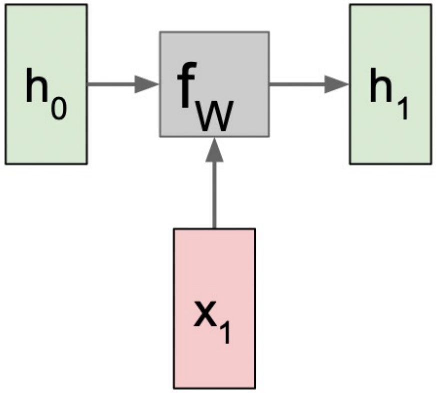
We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

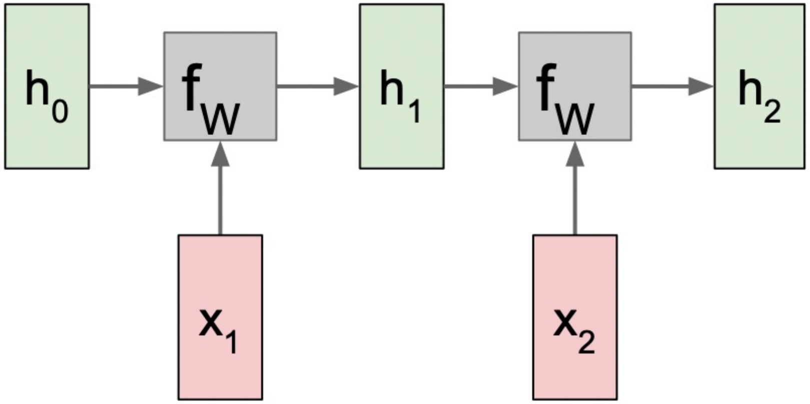
Notice: the same function and the same set of parameters are used at every time step.



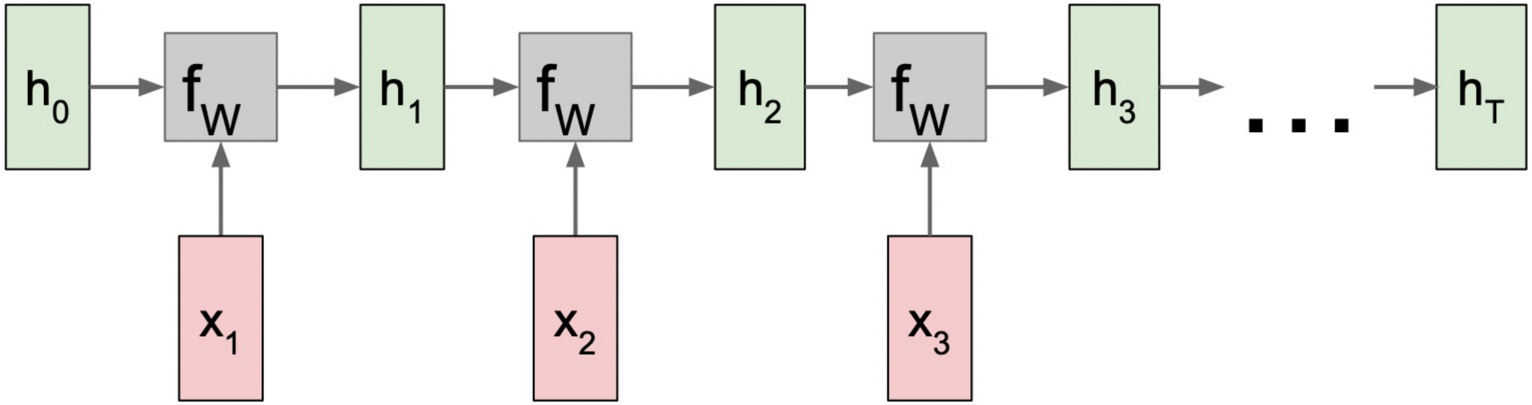
RNN: Computational Graph



RNN: Computational Graph

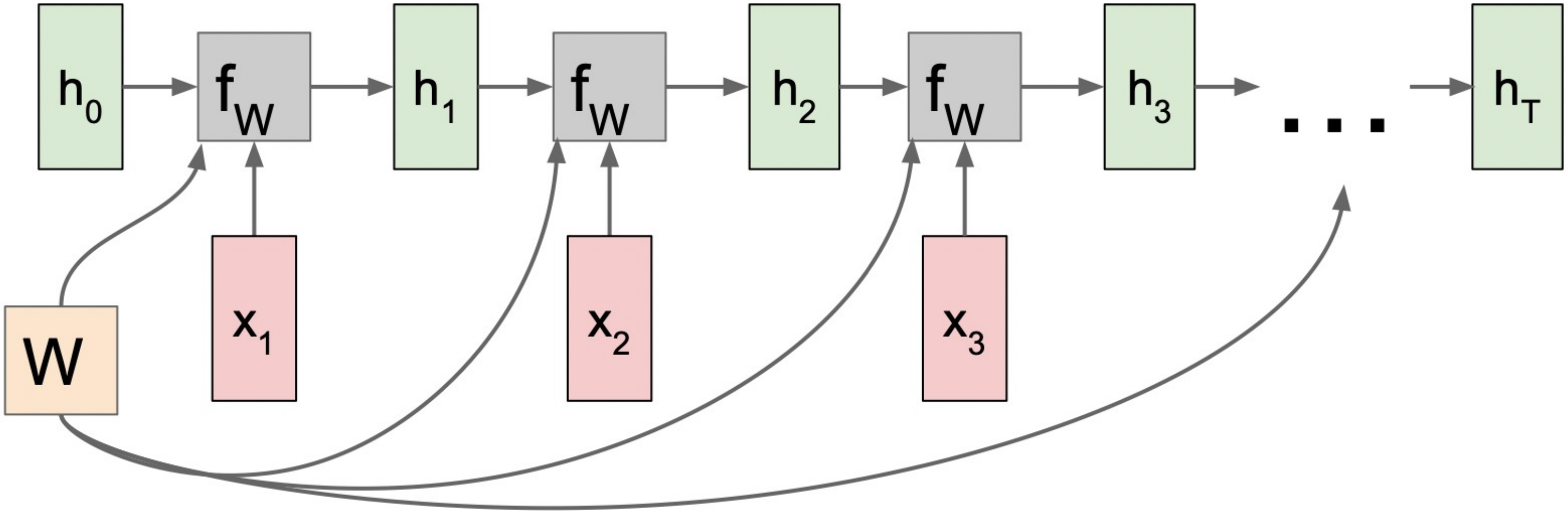


RNN: Computational Graph



RNN: Computational Graph

Re-use the same weight matrix at every time-step



Learning Resources

- Charu C. Aggarwal, Neural Networks and Deep Learning, Springer, 2018.
- Eugene Charniak, Introduction to Deep Learning, MIT Press, 2018.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
- Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.
- Deng & Yu, Deep Learning: Methods and Applications, Now Publishers, 2013.
- http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf
- https://www.youtube.com/watch?v=6niqTuYFZLQ

Thank you