LECTURE 12

RNN

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

LECTURE 12-1

RNN INTRODUCTION

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

Sequence Data

01. We don't understand one word only

02. We understand based on the previous words
+ this word. (time series)

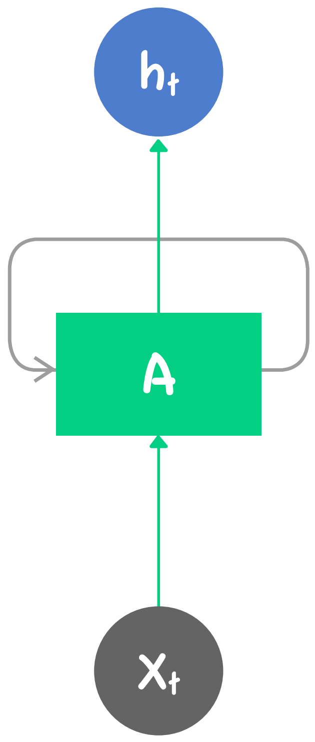
03. NN/CNN cannot do this

Sequence Data

01. We don't understand one word only

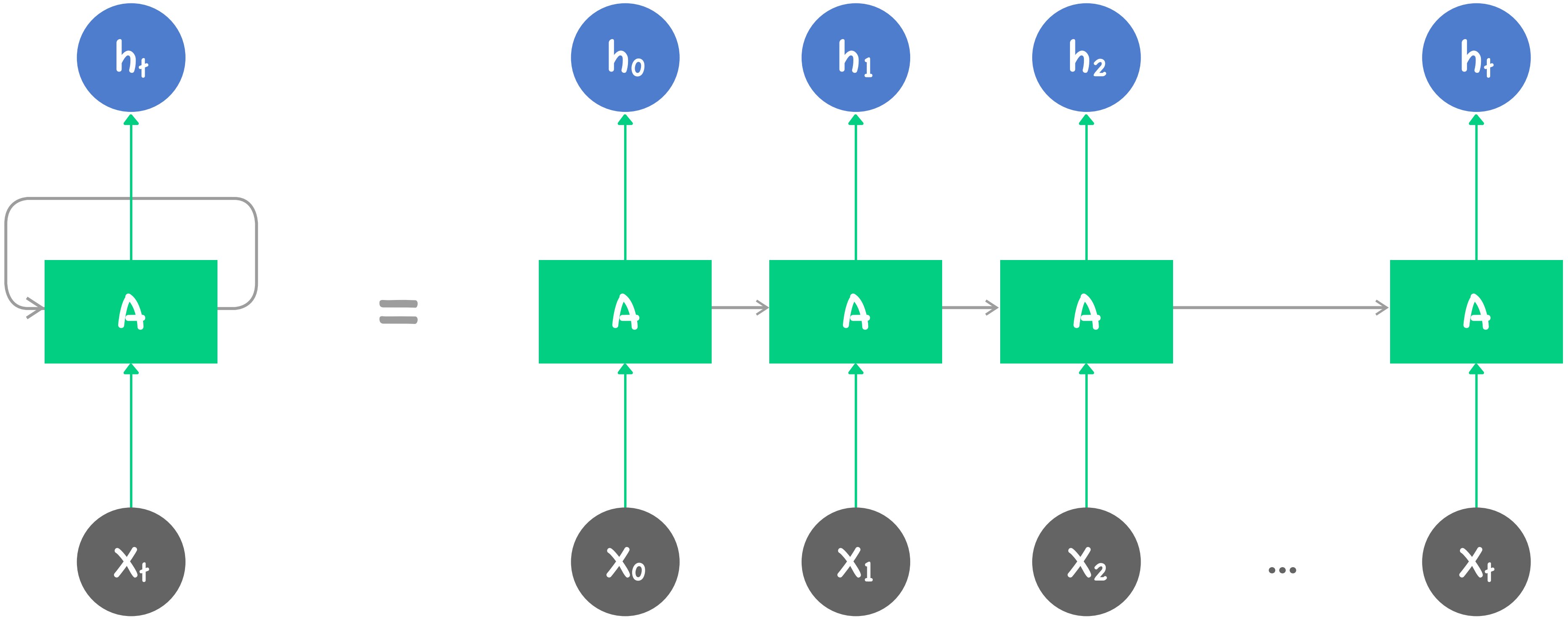
02. We understand based on the previous words + this word. (time series)

03. NN/CNN cannot do this



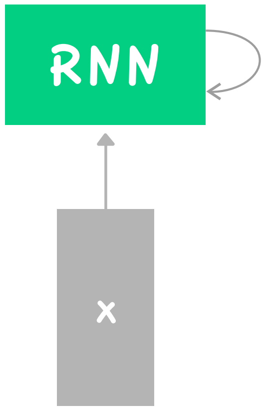
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Sequence Data

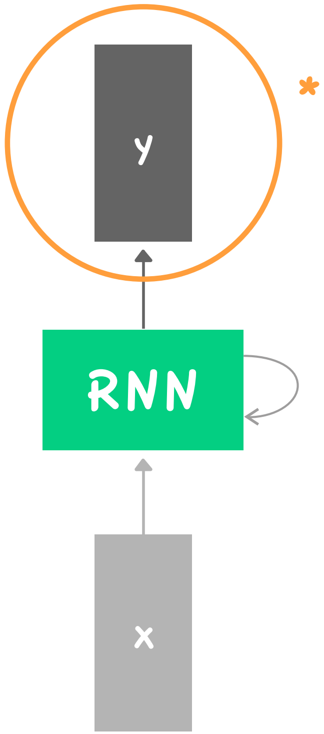


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

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:

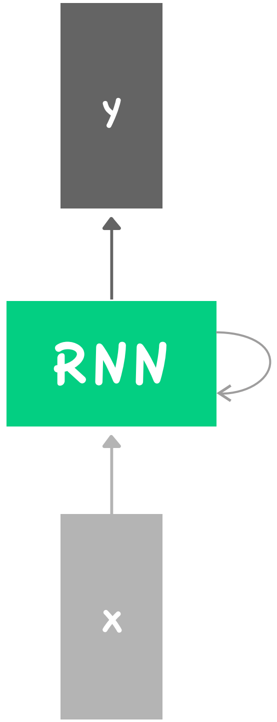
$$h_t = f_W(h_{t-1}, x_t)$$

New state

Some function with parameters W

Old state

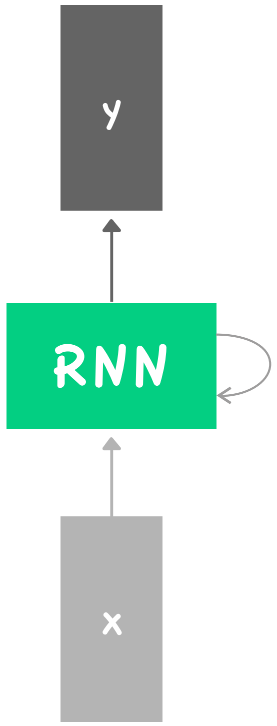
Input vector at some time step



Recurrent Neural Network

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

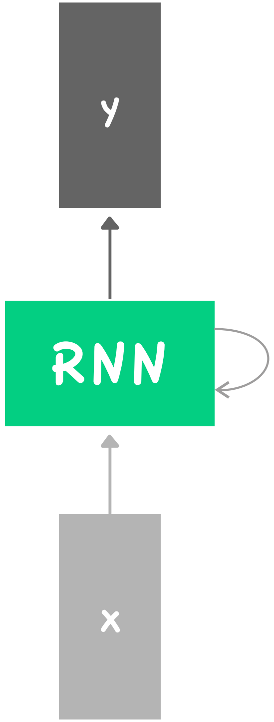
$$h_t = f_W(h_{t-1}, x_t)$$



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

Recurrent Neural Network (Vanilla)

The state consists of a single "hidden" vector **h**:

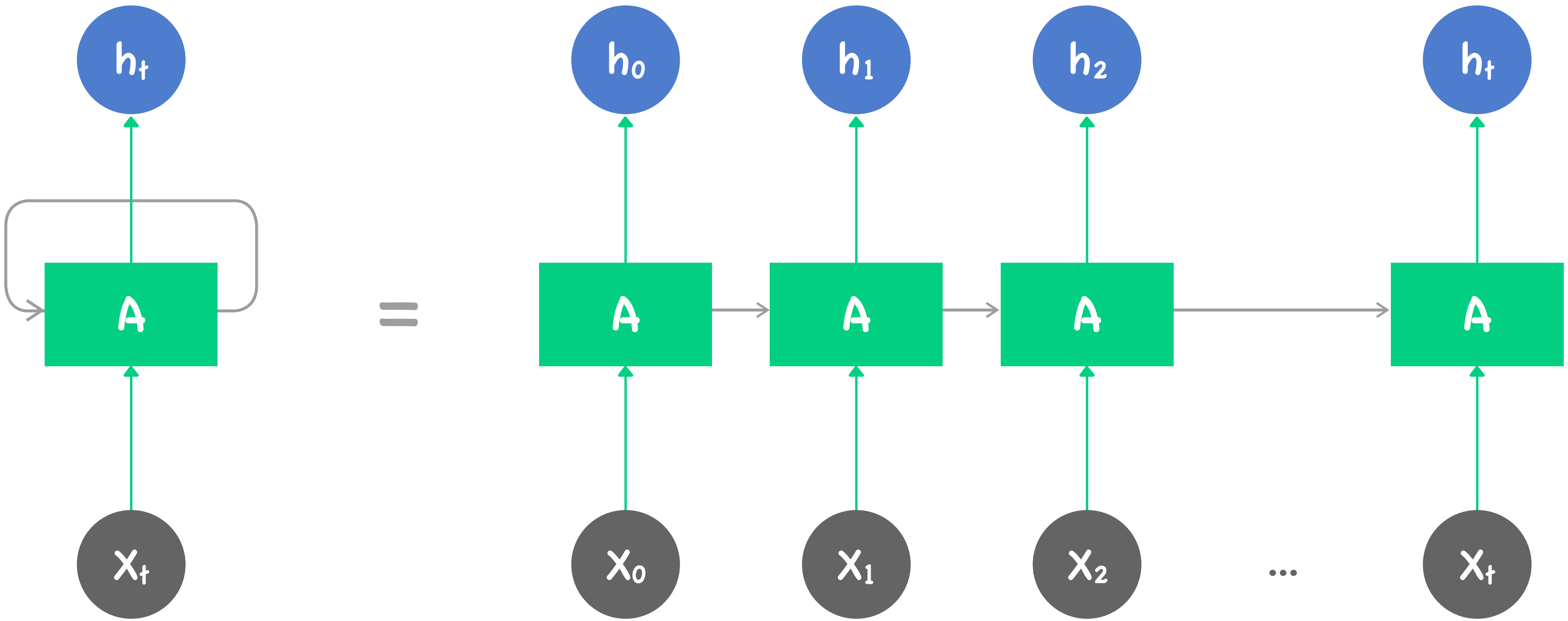


$$h_t = fw(h_{t-1}, x_t)$$

$$\downarrow$$

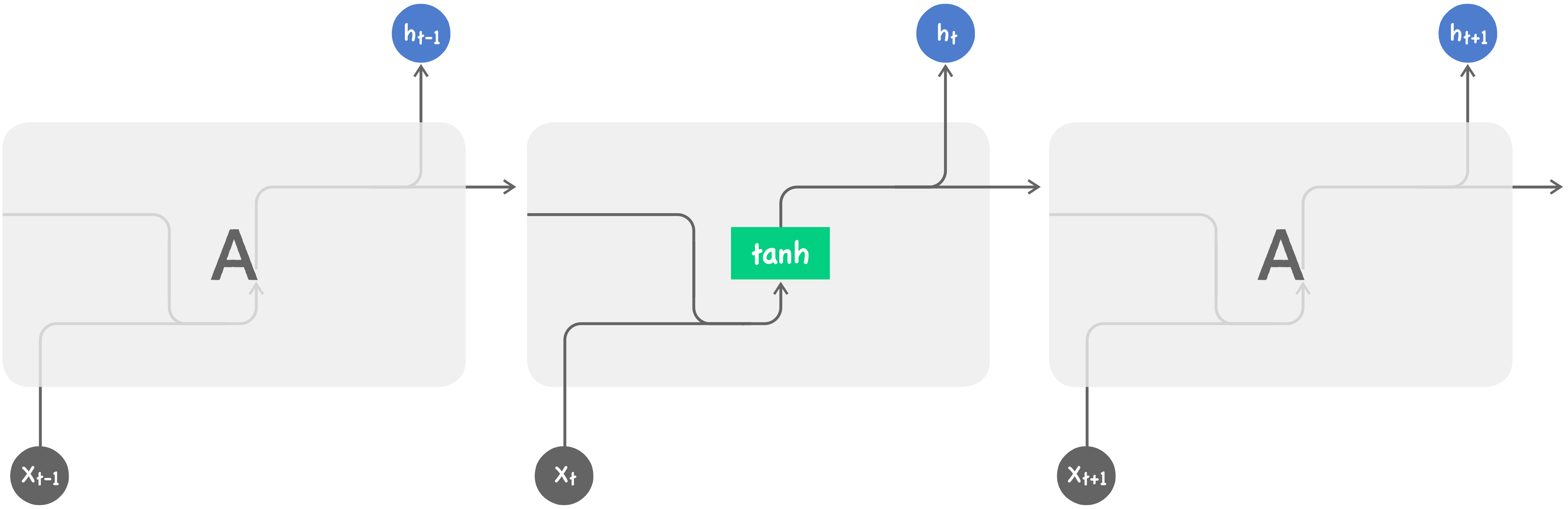
$$h_t = \tanh(W_{hh}h_{t-1}, W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$



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

http://colah.github.io/posts/2015-08-Understanding-LSTMs/



Given list of word vectors : $x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$

At a single time step : $h_t = \sigma(W^{(hh)}h_{t-1} + W^{(hx)}x_{[t]})$

$$\hat{y}_t = \mathrm{softmax}(W^{(S)}h_t)$$

$$\hat{P}(x_{t+1} = v_j \mid x_t, \ldots, x_1) = \hat{y}_{t,j}$$

Character-level Language Model

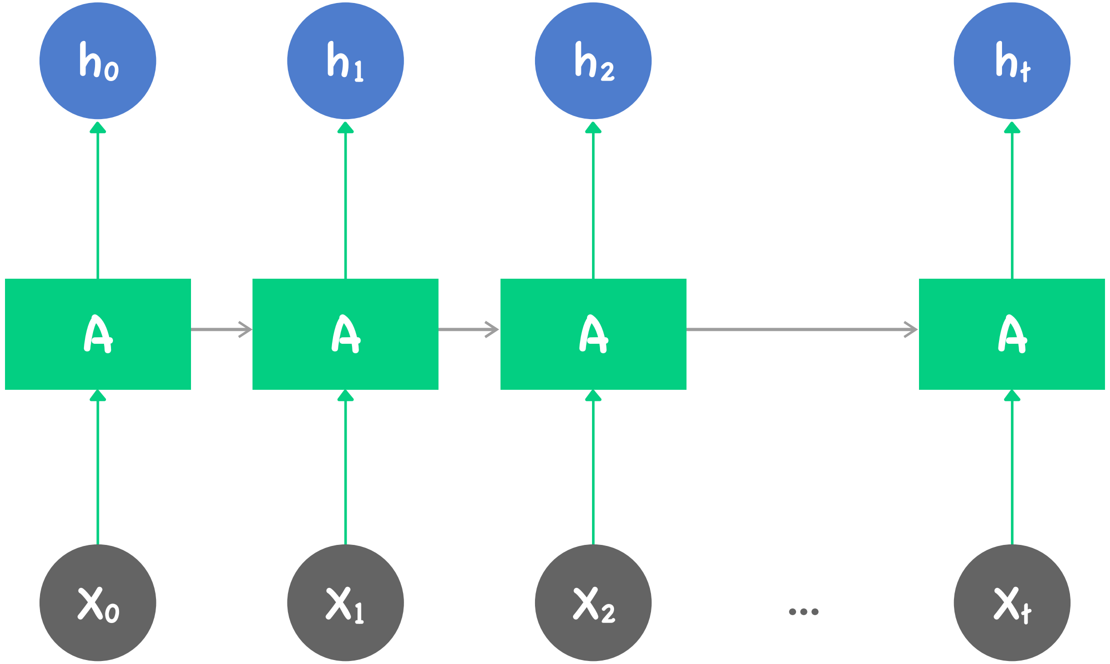
Example

Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"



Character-level Language Model

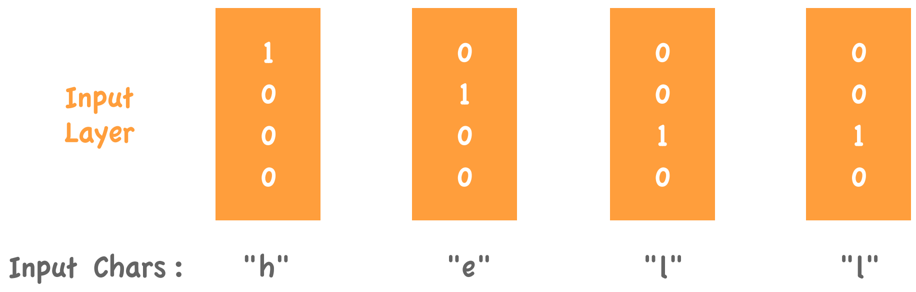
Example

Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"



Character-level Language Model

Example

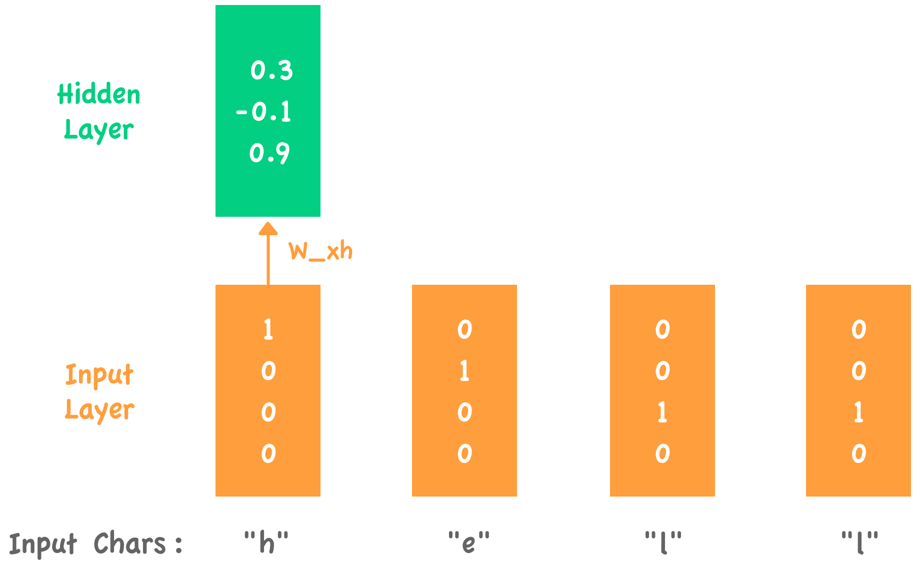
Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Character-level Language Model

Example

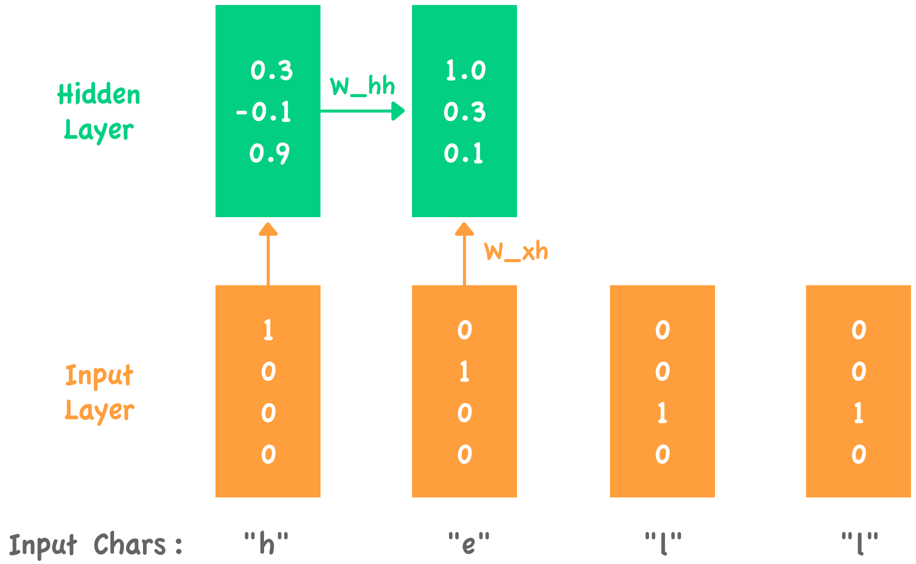
Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Character-level Language Model

Example

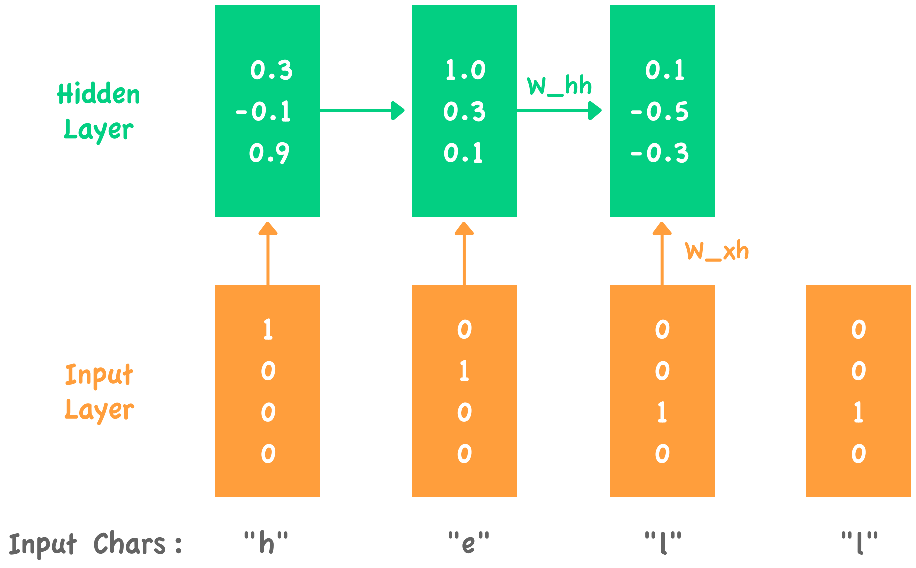
Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Character-level Language Model

Example

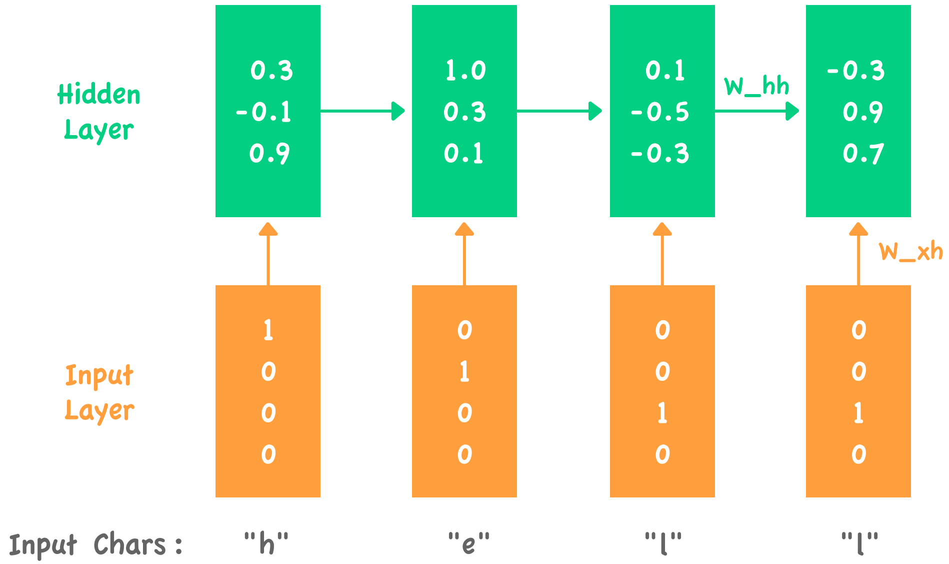
Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"

$$\boxed{h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)}$$



Character-level Language Model

Example

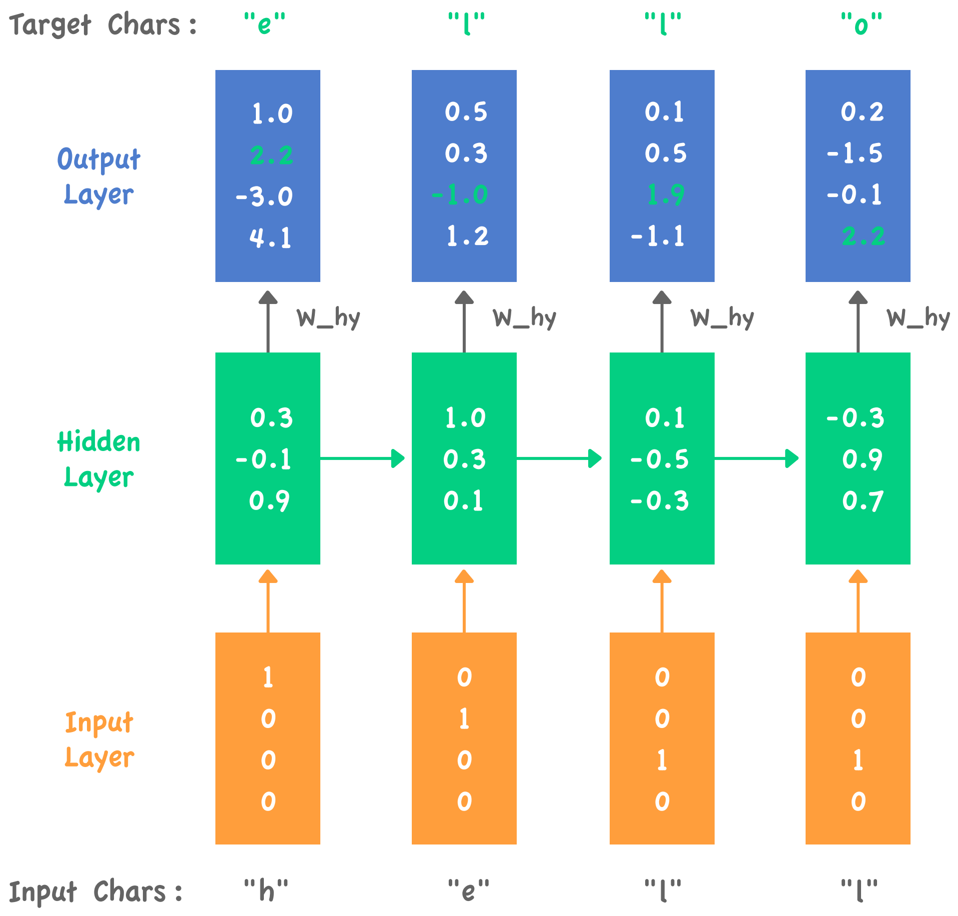
Vocabulary

[h,e,l,o]

Example Training Sequence :

"hello"

$$y_t = W_{hy} h_t$$



Character-level Language Model

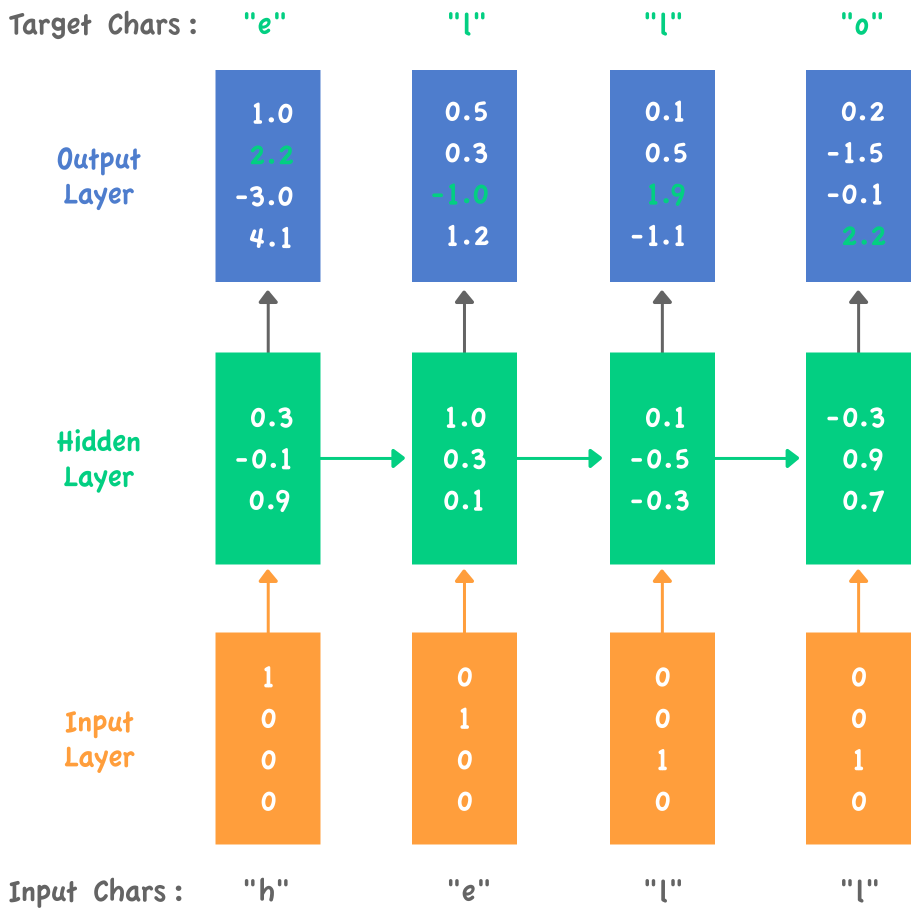
Example

Vocabulary

[h,e,l,o]

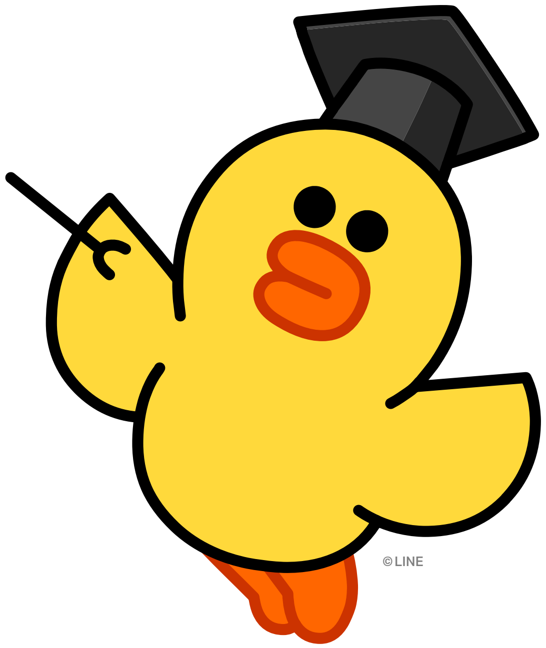
Example Training Sequence :

"hello"



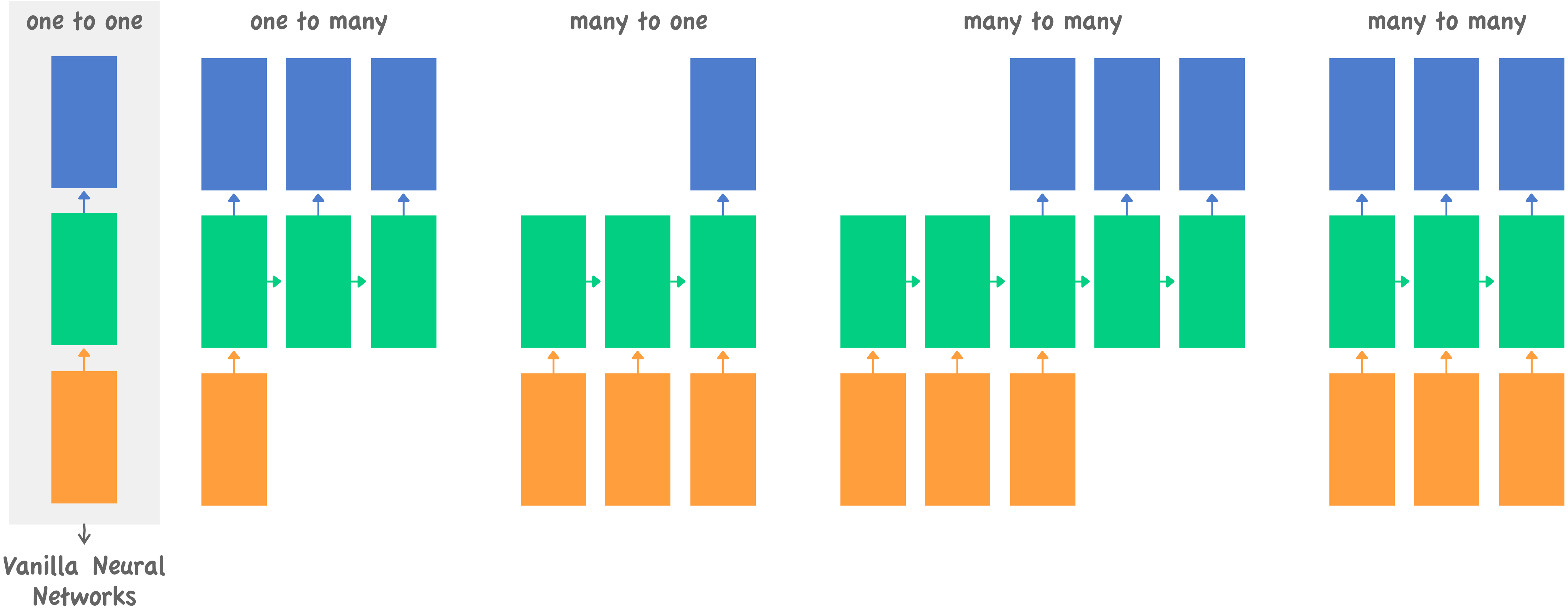
RNN Applications

- Language Modeling
- Speech Recognition
- Machine Translation
- Conversation Modeling / Question Answering
- Image / Video Captioning
- Image / Music / Dance Generation

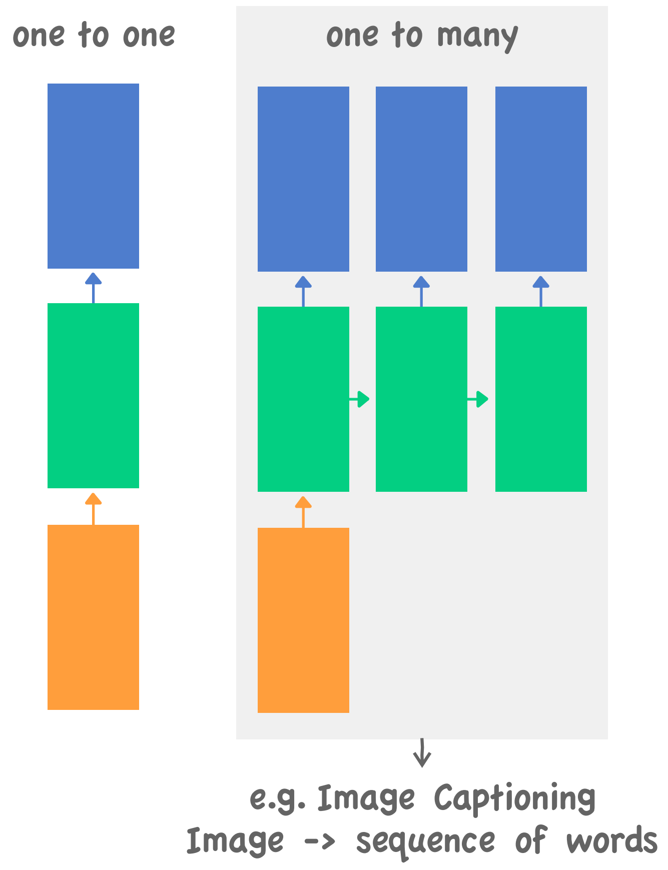


http://jiwonkim.org/awesome-rnn/
https://github.com/TensorFlowKR/awesome_tensorflow_implementations

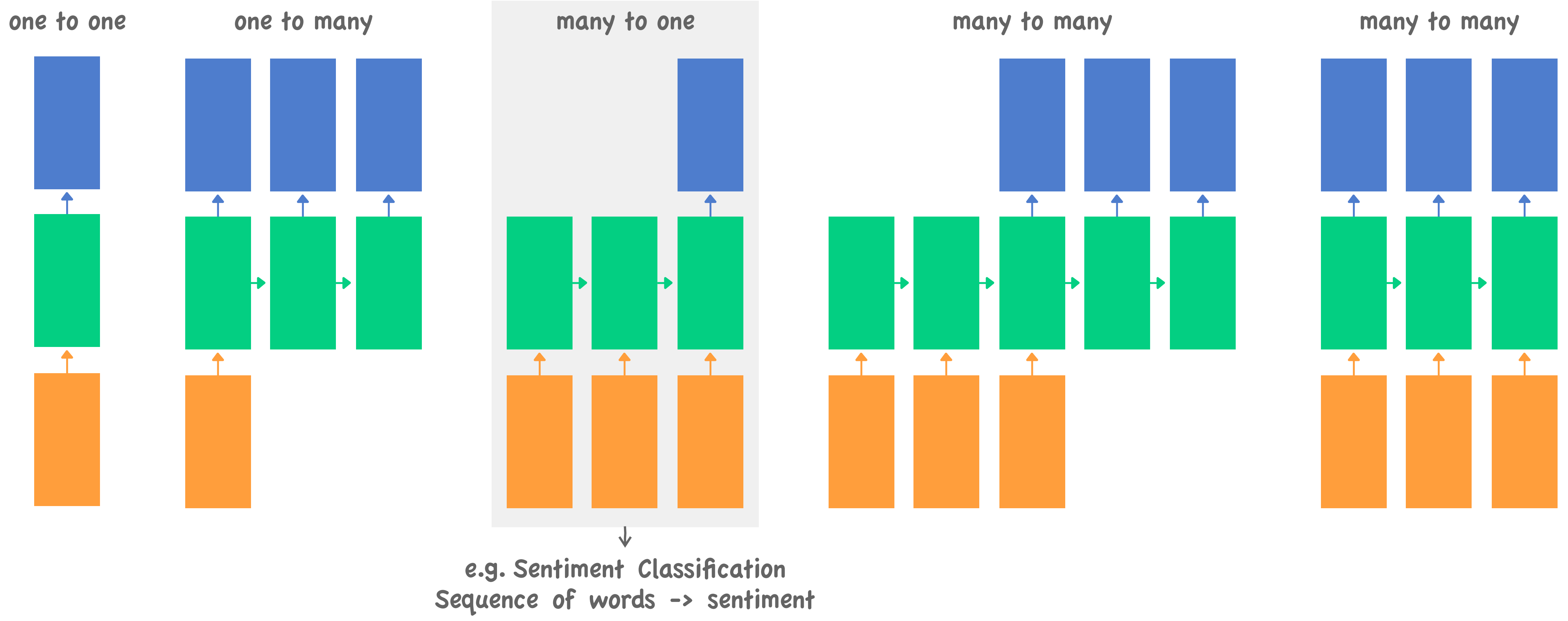
Recurrent Networks Offer a Lot of Flexibility



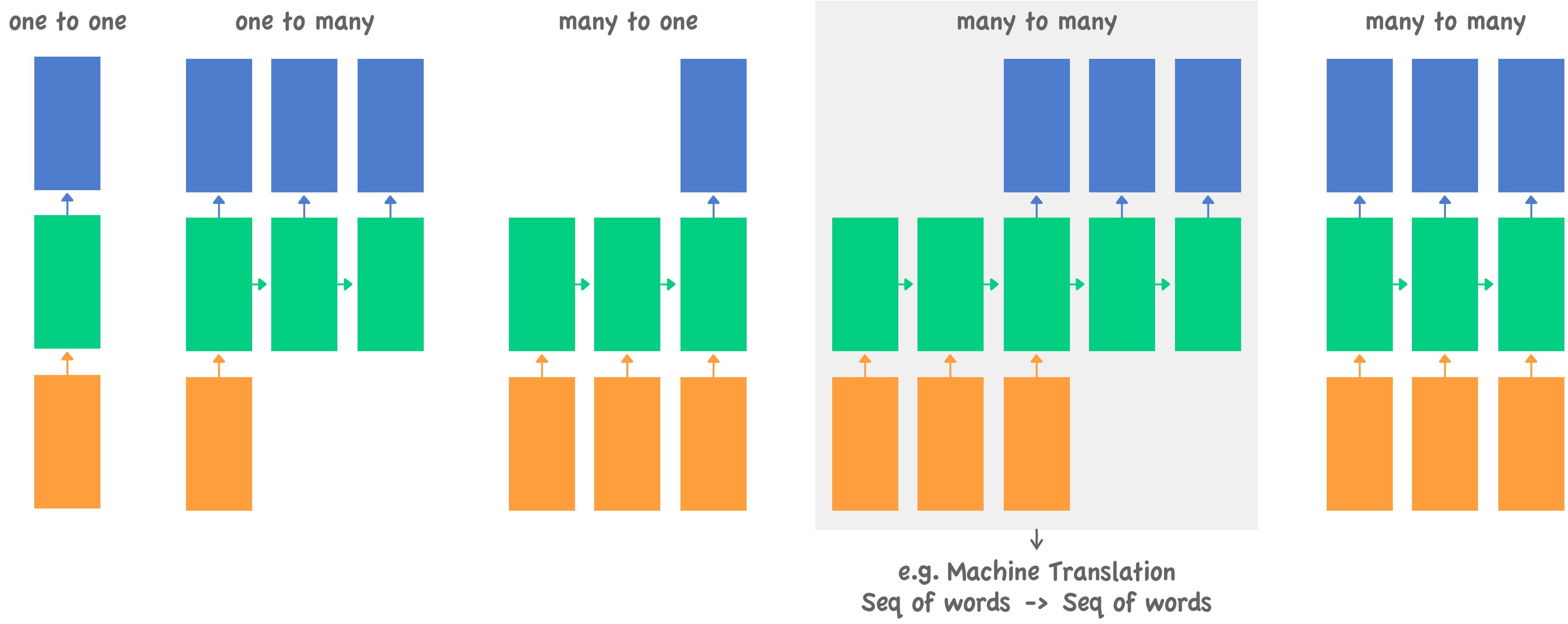
Recurrent Networks Offer a Lot of Flexibility



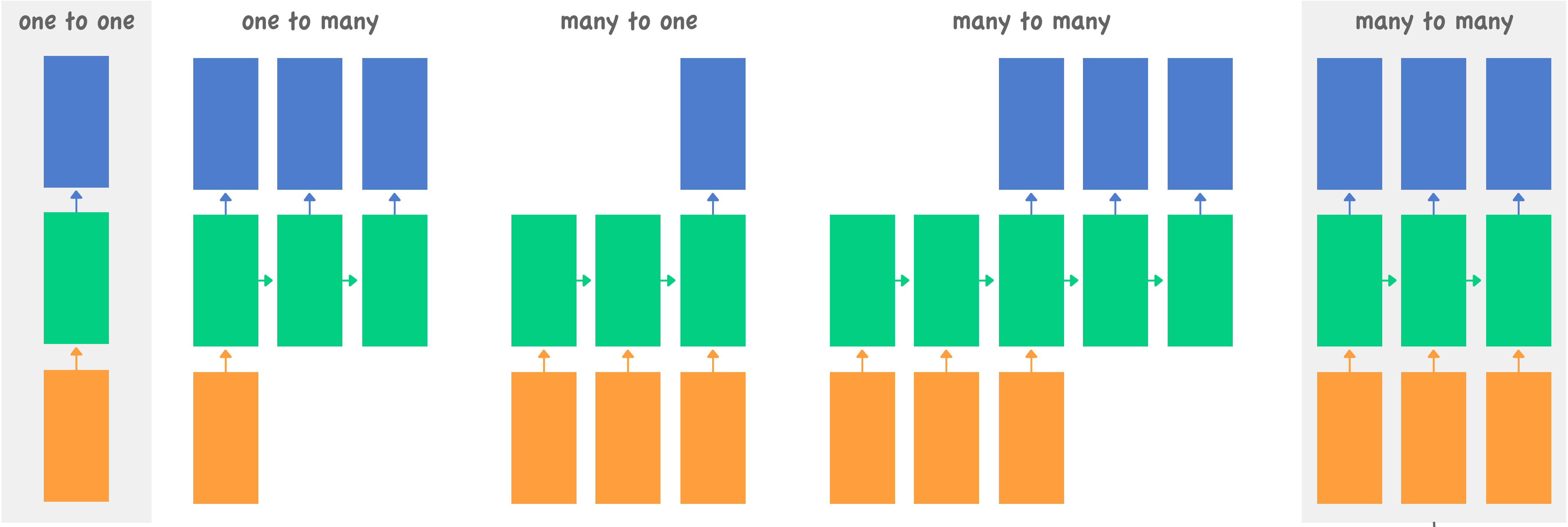
Recurrent Networks Offer a Lot of Flexibility



Recurrent Networks Offer a Lot of Flexibility

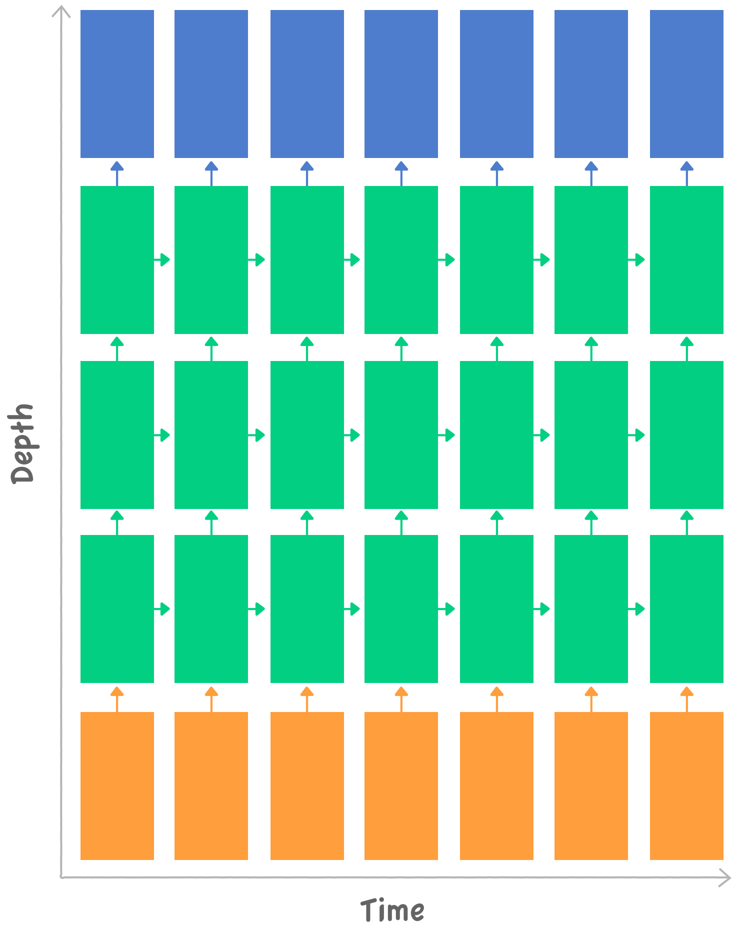


Recurrent Networks Offer a Lot of Flexibility



e.g. Video classification
on frame level

Multi-Layer RNN



Training RNNs Is Challenging

- Several advanced models
  - Long Short Term Memory (LSTM)
  - GRU by Cho et al. 2014

NEXT LECTURE

LSTM INTRODUCTION