Lecture 14

Introduction to Recurrent Neural Networks (Part 1)

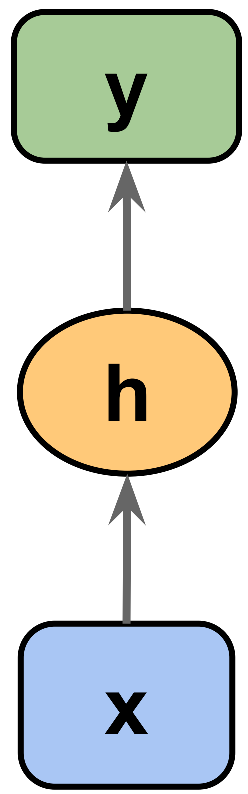
STAT 479: Deep Learning, Spring 2019

Sebastian Raschka

http://stat.wisc.edu/~sraschka/teaching/stat479-ss2019/

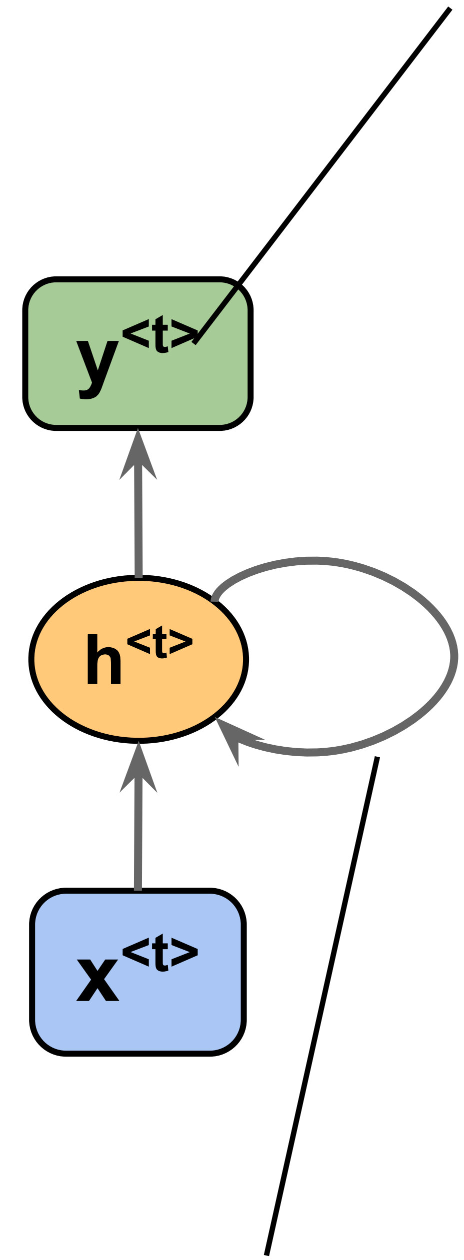
Overview

Networks we used previously: also called feedforward neural networks



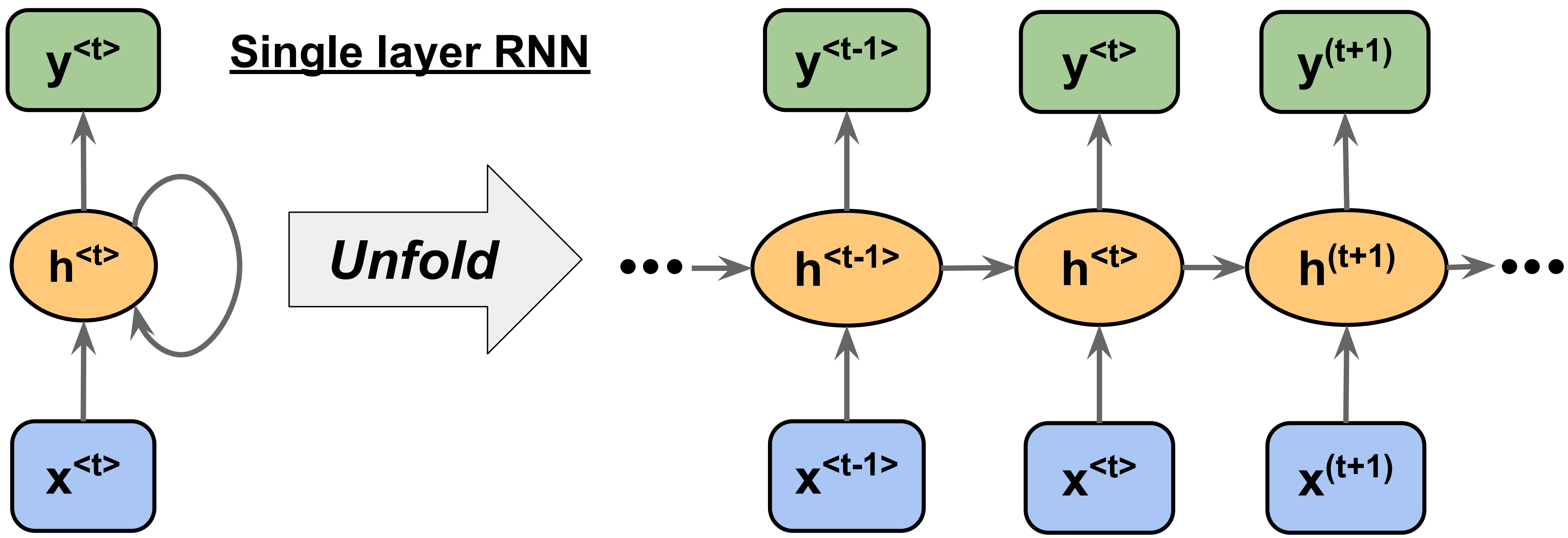
Recurrent Neural Network (RNN)

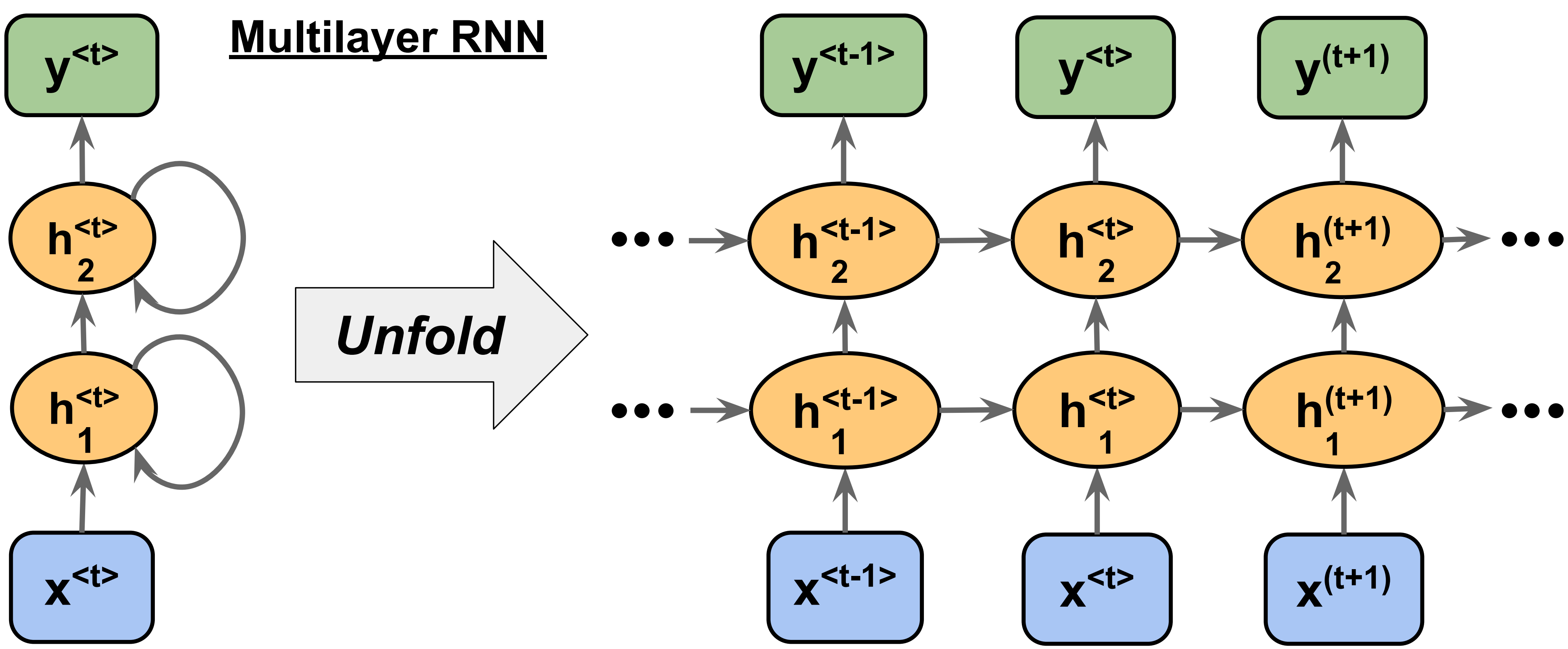
time step t



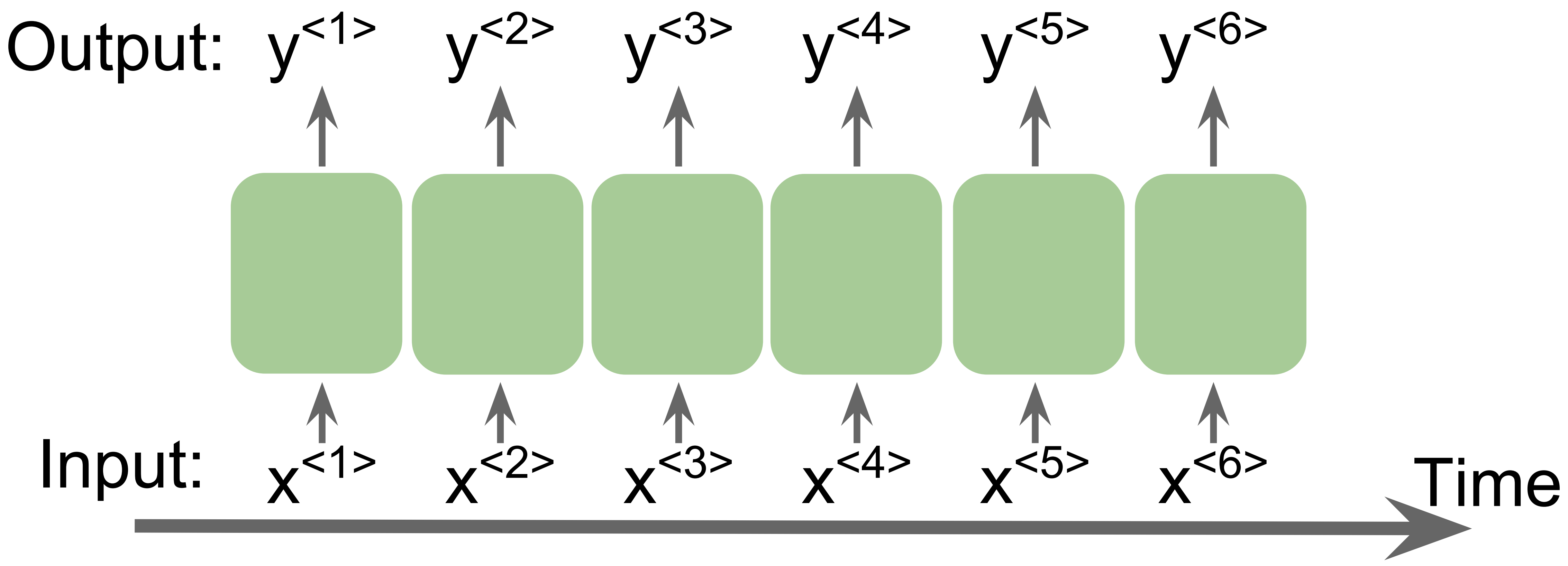
Recurrent edge

Overview


Single layer RNN
y<t>
h<t>
x<t>
Unfold
y<t-1>
y<t>
y(t+1)
h<t-1>
h<t>
h(t+1)
x<t-1>
x<t>
x(t+1)


Multilayer RNN
y<t>
h<t> 2
h<t> 1
x<t>
Unfold
y<t-1>
y<t>
y(t+1)
h<t-1> 2
h<t> 2
h(t+1) 2
h<t-1> 1
h<t> 1
h(t+1) 1
x<t-1>
x<t>
x(t+1)

Sequential data is not i.i.d.



Applications: Working with Sequential Data

- Text classification
- Speech recognition (acoustic modeling)
- language translation
- ...

Stock market predictions

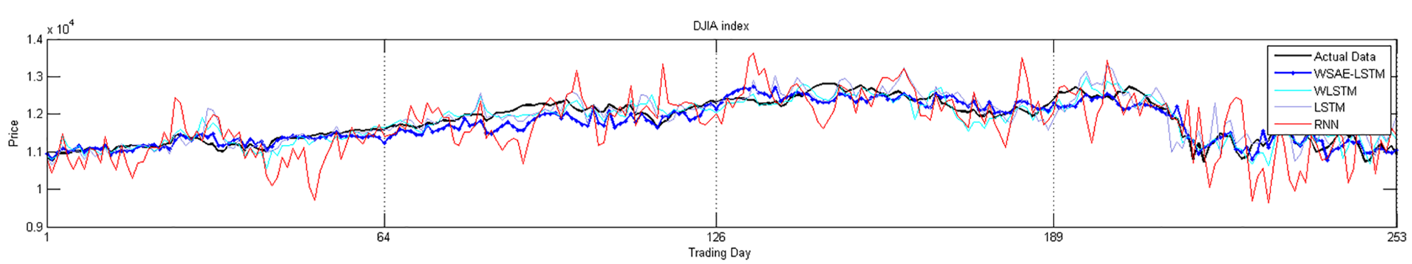
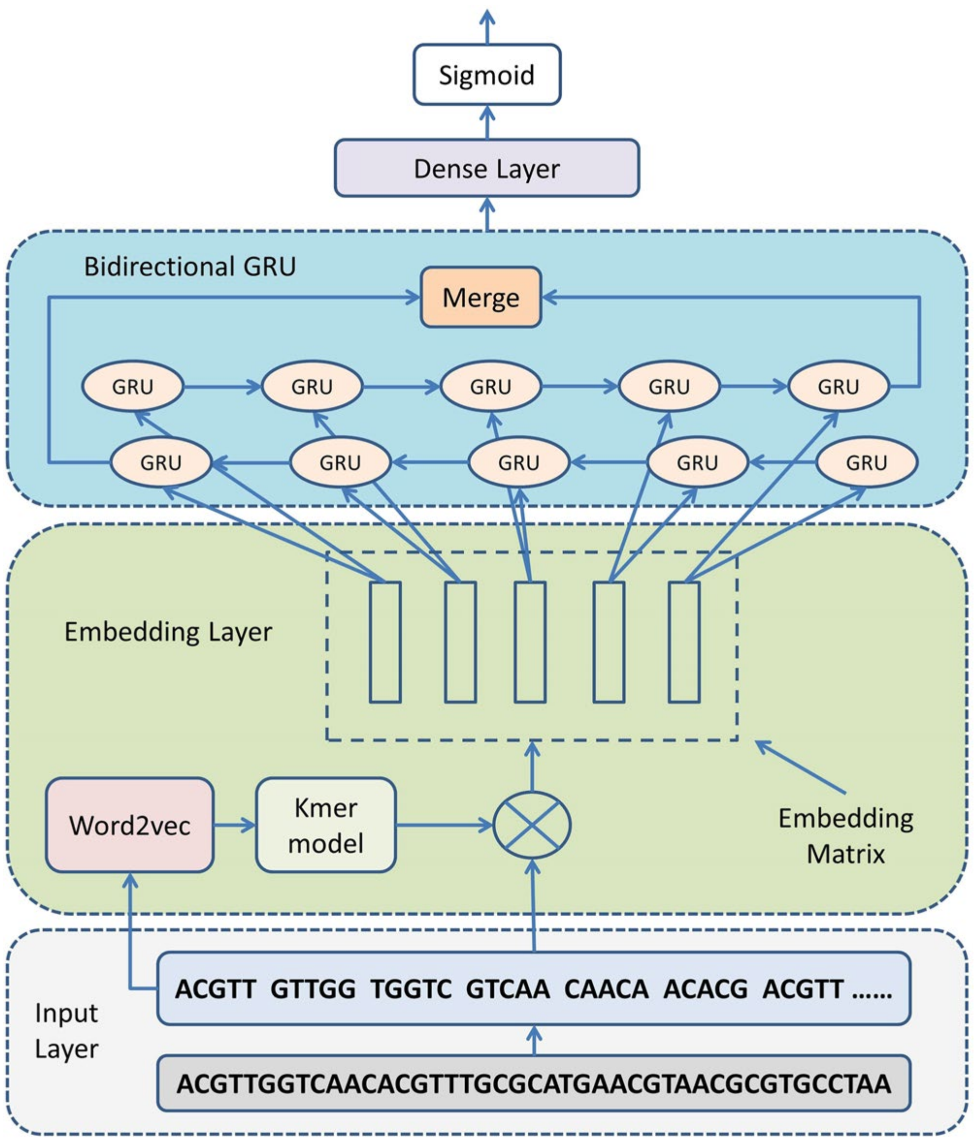


Fig 8. Displays the actual data and the predicted data from the four models for each stock index in Year 1 from 2010.10.01 to 2011.09.30.

https://doi.org/10.1371/journal.pone.0180944.g008

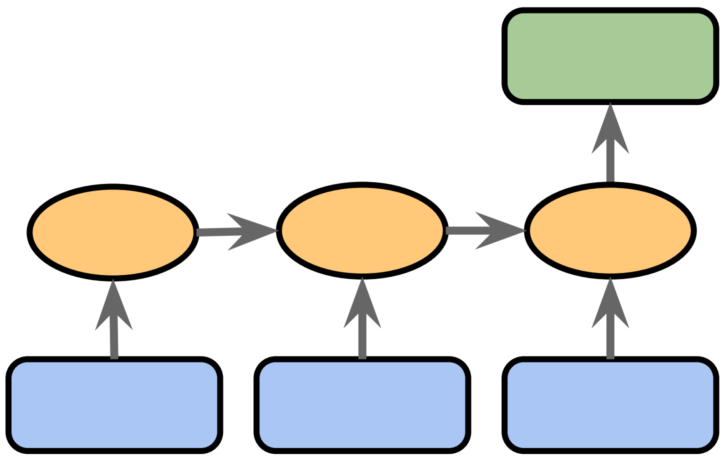
Bao, Wei, Jun Yue, and Yulei Rao. "A deep learning framework for financial time series using stacked autoencoders and long-short term memory." *PloS one* 12, no. 7 (2017): e0180944.



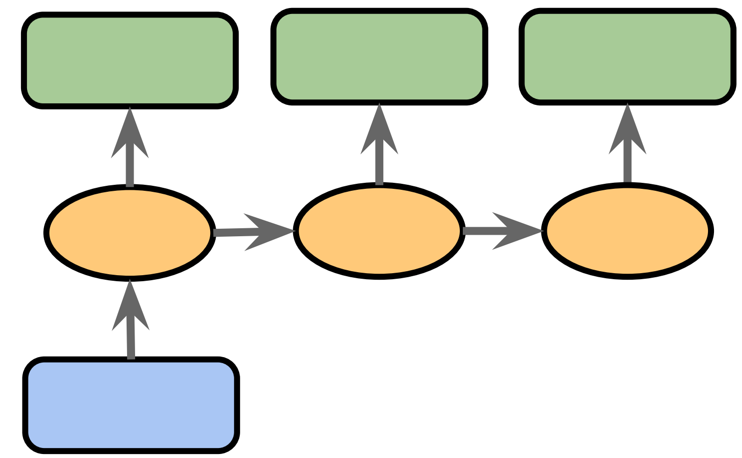
Shen, Zhen, Wenzheng Bao, and De-Shuang Huang. "Recurrent Neural Network for Predicting Transcription Factor Binding Sites." *Scientific reports* 8, no. 1 (2018): 15270.

DNA or (amino acid/protein) sequence modeling

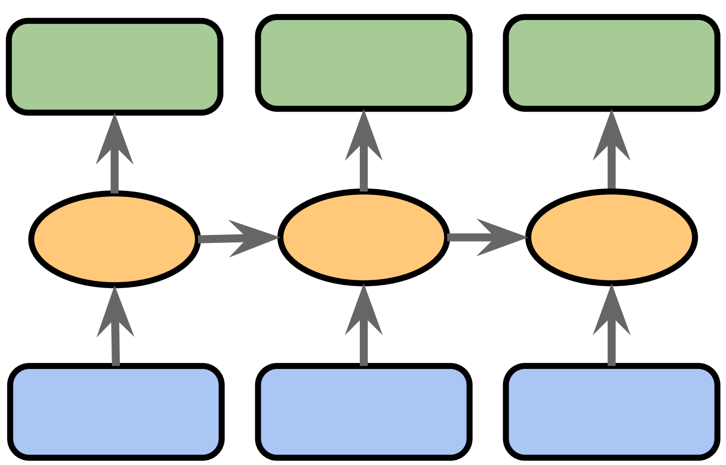
Different Types of Sequence Modeling Tasks



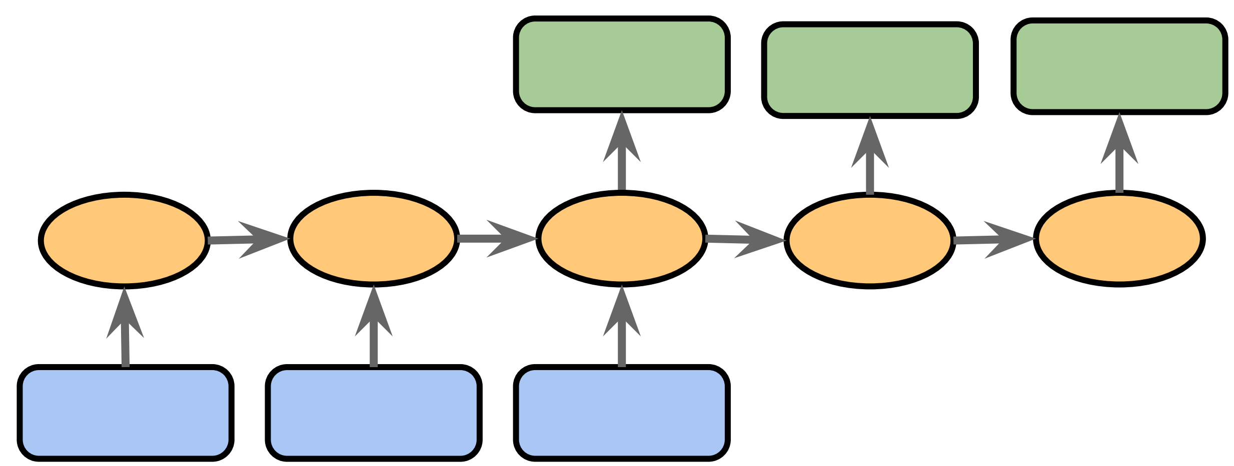
many-to-one



one-to-many



many-to-many

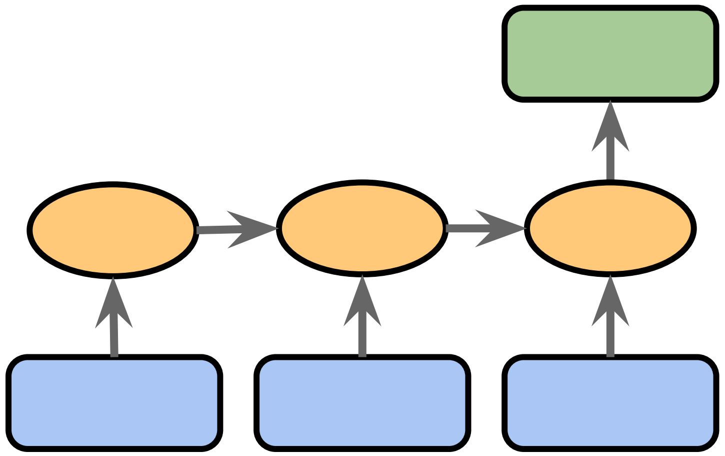


many-to-many

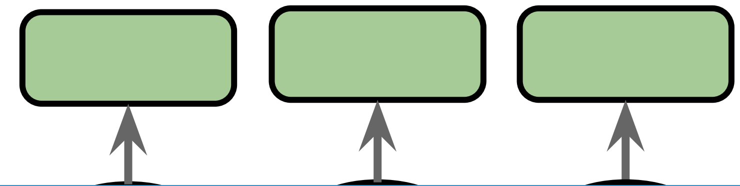
Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Different Types of Sequence Modeling Tasks

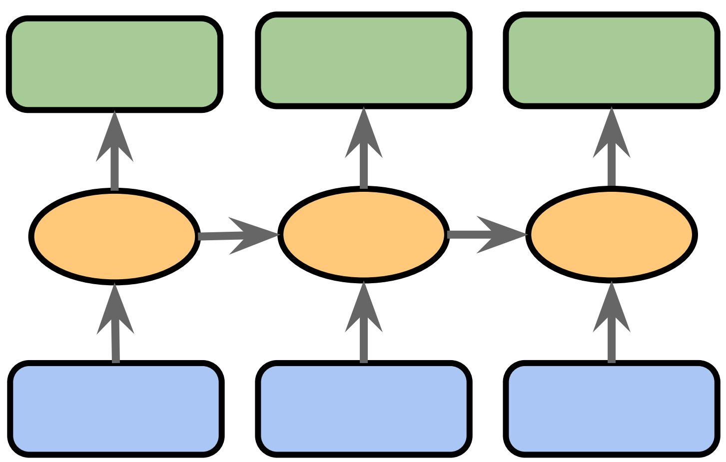


many-to-one



Many-to-one: The input data is a sequence, but the output is a fixed-size vector, not a sequence.

Ex.: sentiment analysis, the input is some text, and the output is a class label.



many-to-many

many-to-many

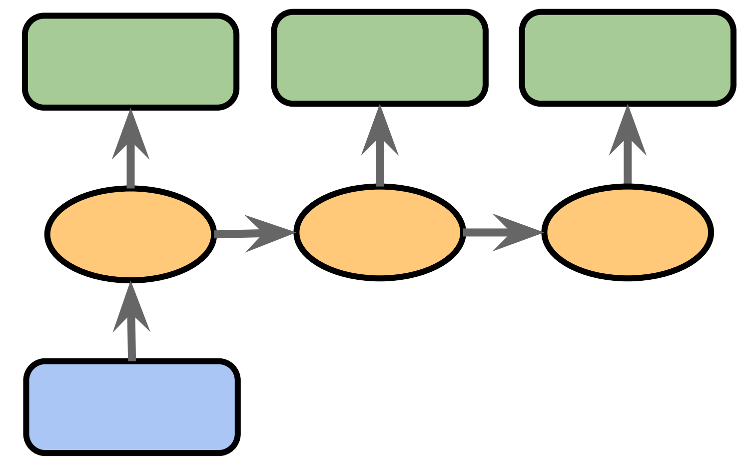
Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

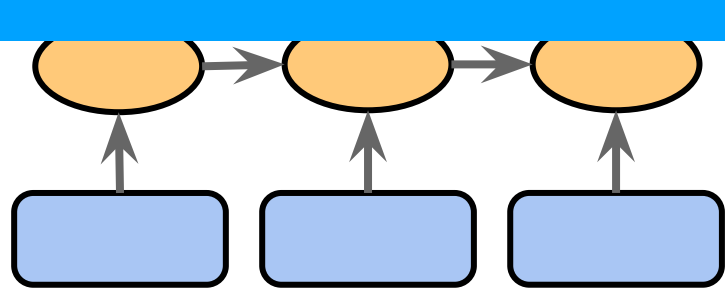
Different Types of Sequence Modeling Tasks

One-to-many: Input data is in a standard format (not a sequence), the output is a sequence.

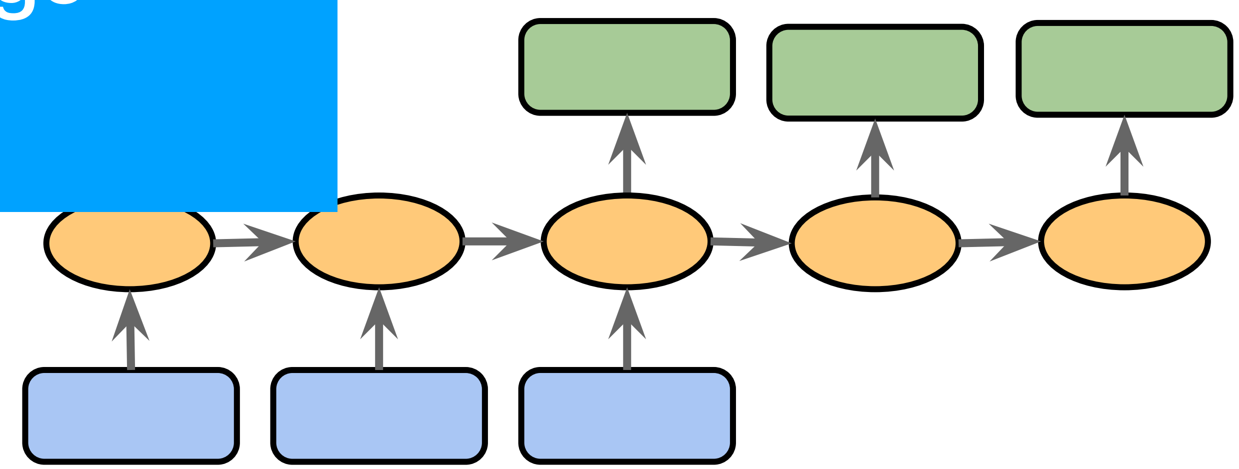
Ex.: Image captioning, where the input is an image, the output is a text description of that image



one-to-many



many-to-many



many-to-many

Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

Different Types of Sequence Modeling Tasks

Many-to-many: Both inputs and outputs are sequences.
Can be direct or delayed.

Ex.: Video-captioning, i.e., describing a sequence of images via text (direct).
Translating one language into another (delayed)

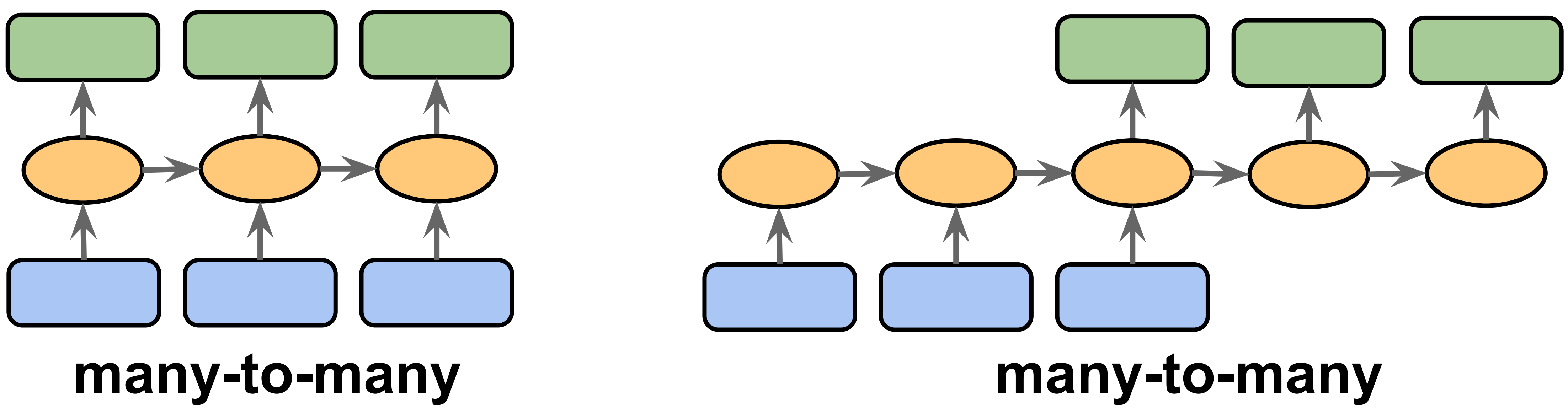


Figure based on:

The Unreasonable Effectiveness of Recurrent Neural Networks by Andrej Karpathy (http://karpathy.github.io/2015/05/21/rnn-effectiveness/)

The Classic Text Classification Approach

A Classic Approach for Text Classification: Bag of Words Model

1) Suppose you want to design a classifier and you have a training dataset consisting of 3 examples (sentences)

$$\mathbf{x}^{[1]} = \text{"The sun is shining"}$$

$$\mathbf{x}^{[2]} = \text{"The weather is sweet"}$$

$$\mathbf{x}^{[3]} = \text{"The sun is shining,} \\ \text{the weather is sweet, and one and one is two"}$$

A Classic Approach for Text Classification: Bag of Words Model

2) Based on ALL your data, you would construct a <u>vocabulary</u> of all unique words

$\mathbf{x}^{[1]} =$ ”The sun is shining”

$\mathbf{x}^{[2]} =$ ”The weather is sweet”

$\mathbf{x}^{[3]} =$ ”The sun is shining, the weather is sweet, and one and one is two”

vocabulary = {
'and': 0,
'is': 1
'one': 2,
'shining': 3,
'sun': 4,
'sweet': 5,
'the': 6,
'two': 7,
'weather': 8,
}

A Classic Approach for Text Classification: Bag of Words Model

3) Use the vocabulary to transform the dataset into bag-of-words vectors
(vector size is determined by the vocabulary size)

$\mathbf{x}^{[1]} =$ "The sun is shining"

$\mathbf{x}^{[2]} =$ "The weather is sweet"

$\mathbf{x}^{[3]} =$ "The sun is shining, the weather is sweet, and one an

vocabulary = {
'and': 0,
'is': 1
'one': 2,
'shining': 3,
'sun': 4,
'sweet': 5,
'the': 6,
'two': 7,
'weather': 8,
}

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

A Classic Approach for Text Classification: Bag of Words Model

4) Use the bag-of-words representation to fit a predictive model (logistic regression, multilayer-perceptron, etc.)

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix} \xrightarrow{\text{train}} \text{Classifier}$$

$$\mathbf{y} = [0, 1, 0]$$

A Classic Approach for Text Classification: Bag of Words Model

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

Rows are training examples

Columns are features

Features can be

- word counts / term frequencies (how often a word appears in the sentence, like above)
- binary 0/1 (whether a word occurs or not)
- term frequency-inverse document frequencies (normalized word counts)
- ...

A Classic Approach for Text Classification: Bag of Words Model

Optional Preprocessing: Stop Word Removal

$\mathbf{x}^{[1]} =$ "~~The~~ sun ~~is~~ shining"

$\mathbf{x}^{[2]} =$ "~~The~~ weather ~~is~~ sweet"

$\mathbf{x}^{[3]} =$ "~~The~~ sun ~~is~~ shining,
~~the~~ weather ~~is~~ sweet, ~~and~~ one ~~and~~ one ~~is~~ two"

A Classic Approach for Text Classification: Bag of Words Model

Optional Preprocessing: n-gram tokenization with n > 1

1 token = 1 word:

$$\mathbf{x}^{[1]} = \text{"}\boxed{\text{The}}\,\boxed{\text{sun}}\,\boxed{\text{is}}\,\boxed{\text{shining}}\text{"}$$

1 token = 2 words:

$$\mathbf{x}^{[1]} = \text{"The sun is shining"}$$

A Classic Approach for Text Classification: Bag of Words Model

For a self-contained example of this "classic" approach, see

https://github.com/rasbt/python-machine-learning-book-2nd-edition/blob/master/code/ch08/ch08.ipynb

A Classic Approach for Text Classification: Bag of Words Model

Big Downside: We lose the spatial relationship between words!

$\mathbf{x}^{[1]} =$ "The sun is shining"

$\mathbf{x}^{[2]} =$ "The weather is sweet"

$\mathbf{x}^{[3]} =$ "The sun is shining, the weather is sweet, and one an

```
vocabulary = {
    'and': 0,
    'is': 1
    'one': 2,
    'shining': 3,
    'sun': 4,
    'sweet': 5,
    'the': 6,
    'two': 7,
    'weather': 8,
}
```

$$\mathbf{X} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \end{bmatrix}$$

Recurrent Neural Networks
(to be continued ...)