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)

Recurrent edge
Overview

Single layer RNN

Multilayer RNN
Sequential data is not i.i.d.

Output: $y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

Input: $x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$ $x^{<6>}$
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


Different Types of Sequence Modeling Tasks

- **many-to-one**
- **one-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: 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.

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.

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)

\( x^{[1]} = \text{"The sun is shining"} \)
\( x^{[2]} = \text{"The weather is sweet"} \)
\( x^{[3]} = \text{"The sun is shining, the weather is sweet, and one and one is two"} \)
A Classic Approach for Text Classification: 
Bag of Words Model

2) Based on ALL your data, you would construct a **vocabulary** of all unique words

\[
\begin{align*}
\text{x}^{[1]} & = \text{"The sun is shining"} \\
\text{x}^{[2]} & = \text{"The weather is sweet"} \\
\text{x}^{[3]} & = \text{"The sun is shining, the weather is sweet, and one and one is two"}
\end{align*}
\]

\[
\text{vocabulary} = \{
\begin{align*}
\text{\'and\': 0,} \\
\text{\'is\': 1} \\
\text{\'one\': 2,} \\
\text{\'shining\': 3,} \\
\text{\'sun\': 4,} \\
\text{\'sweet\': 5,} \\
\text{\'the\': 6,} \\
\text{\'two\': 7,} \\
\text{\'weather\': 8,}
\end{align*}
\}
\]
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)

\[
x^{[1]} = \text{"The sun is shining"}
\]
\[
x^{[2]} = \text{"The weather is sweet"}
\]
\[
x^{[3]} = \text{"The sun is shining, the weather is sweet, and one and}
\]

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

\[
X = \begin{bmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \\
\end{bmatrix}
\]
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.)

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

\[\mathbf{y} = [0, 1, 0]\]
A Classic Approach for Text Classification:
Bag of Words Model

\[ X = \begin{bmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1
\end{bmatrix} \]

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

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

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

\[
x^{[3]} = "The sun is shining, the weather is sweet, and one and one is two"
\]
A Classic Approach for Text Classification: Bag of Words Model

Optional Preprocessing: n-gram tokenization with n > 1

1 token = 1 word:

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

1 token = 2 words:

\[ x^{[1]} = "\text{The sun is shining}" \]
A Classic Approach for Text Classification: Bag of Words Model

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

A Classic Approach for Text Classification: Bag of Words Model

Big Downside: We lose the spatial relationship between words!

\[
\begin{align*}
x^1 &= \text{"The sun is shining"} \\
x^2 &= \text{"The weather is sweet"} \\
x^3 &= \text{"The sun is shining, the weather is sweet, and one and two"}
\end{align*}
\]

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

\[
X = \begin{bmatrix}
0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
2 & 3 & 2 & 1 & 1 & 1 & 2 & 1 & 1 \\
\end{bmatrix}
\]
Recurrent Neural Networks
(to be continued ... )