Fundamentals of Deep Learning

ELL 881 Lec 06A: Recurrent Neural Networks

Vineet Kumar

September 12, 2018

Sequential Data

Recurrent Neural Network (RNN) is a family of neural networks for processing sequential data

[1] Borrowed from https://www.cs.toronto.edu/ hinton/csc2515/notes/lec9timeseries.pdf

Sequential Data

Recurrent Neural Network (RNN) is a family of neural networks for processing sequential data

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(\tau)}$$

[1] Borrowed from https://www.cs.toronto.edu/ hinton/csc2515/notes/lec9timeseries.pdf

Sequential Data

Recurrent Neural Network (RNN) is a family of neural networks for processing sequential data

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(\tau)}$$

Examples of sequential data?[1]

---

[1] Borrowed from https://www.cs.toronto.edu/ hinton/csc2515/notes/lec9timeseries.pdf

Sequential Data

Recurrent Neural Network (RNN) is a family of neural networks for processing sequential data

$$\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(\tau)}$$

Examples of sequential data?[1]

- **Ordered Sequence**: Words/Characters in a sentence, Gene sequence
- **Time-series**: Stock Market, Speech, Video analysis

[1] Borrowed from https://www.cs.toronto.edu/ hinton/csc2515/notes/lec9timeseries.pdf

Parameter Sharing

Key Idea Parameter Sharing: Share parameters across different parts of the model.

Parameter Sharing

Key Idea Parameter Sharing: Share parameters across different parts of the model.
Parameter sharing allows to extend and apply the model to examples of different lengths, and generalize!

Parameter Sharing

Key Idea Parameter Sharing: Share parameters across different parts of the model.
Parameter sharing allows to extend and apply the model to examples of different lengths, and generalize!
Without parameter sharing, we would require separate parameters across each time step!

Parameter Sharing

Key Idea Parameter Sharing: Share parameters across different parts of the model.
Parameter sharing allows to extend and apply the model to examples of different lengths, and generalize!
Without parameter sharing, we would require separate parameters across each time step!

In **2009**, I went to Nepal

Parameter Sharing

Key Idea Parameter Sharing: Share parameters across different parts of the model.
Parameter sharing allows to extend and apply the model to examples of different lengths, and generalize!
Without parameter sharing, we would require separate parameters across each time step!

In **2009**, I went to Nepal
I went to Nepal in **2009**

Parameter Sharing

Key Idea Parameter Sharing: Share parameters across different parts of the model.
Parameter sharing allows to extend and apply the model to examples of different lengths, and generalize!
Without parameter sharing, we would require separate parameters across each time step!

In **2009**, I went to Nepal
I went to Nepal in **2009**

Dynamical System and Recurrence

Consider the following network:

$$s^{(t)} = f(s^{(t-1)}; \theta)$$

This network is recurrent as computing $s^{(t)}$ requires $s^{(t-1)}$

Question: How would you unfold this graph for $\tau = 3$ time steps, i.e compute $s^{(3)}$?

Dynamical System and Recurrence

Consider the following network:

$$s^{(t)} = f(s^{(t-1)}; \theta)$$

This network is recurrent as computing $s^{(t)}$ requires $s^{(t-1)}$

Question: How would you unfold this graph for $\tau = 3$ time steps, i.e compute $s^{(3)}$?

$$\begin{aligned} s^{(3)} &= f(s^{(2)}; \theta) \\ &= f(f(s^{(1)}; \theta); \theta) \end{aligned}$$

Dynamical System and Recurrence

Consider the following network:

$$s^{(t)} = f(s^{(t-1)}; \theta)$$

This network is recurrent as computing $s^{(t)}$ requires $s^{(t-1)}$

Question: How would you unfold this graph for $\tau = 3$ time steps, i.e compute $s^{(3)}$?

$$\begin{aligned} s^{(3)} &= f(s^{(2)}; \theta) \\ &= f(f(s^{(1)}; \theta); \theta) \end{aligned}$$

Same parameters θ are used across all time steps!

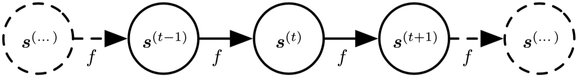

Figure 10.1

Figure: Dynamical System; Image Courtesy: http://www.deeplearningbook.org/slides/10_rnn.pdf

Input Signal Summary

The following system computes $\boldsymbol{h}^{(t)}$, updated by an external signal $\boldsymbol{x}^{(t)}$:

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \theta)$$

Input Signal Summary

The following system computes $\boldsymbol{h}^{(t)}$, updated by an external signal $\boldsymbol{x}^{(t)}$:

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{h}^{(t-1)}, \boldsymbol{x}^{(t)}; \theta)$$

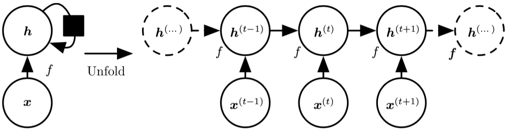

Figure 10.2

Input Signal Summary

$\boldsymbol{h}^{(t)}$ can be thought of as a lossy summary of input seen so far, that is $\boldsymbol{x}^1, \cdots \boldsymbol{x}^t$

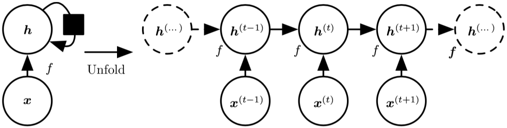

Figure 10.2