Recurrent Neural Networks

Kate Farrahi

ECS Southampton

March 7, 2019

Recurrent Neural Networks - Motivation

x:	Kate	Farrahi	and	Jonathon	Hare	teach	deep	learning
y:	1	1	0	1	1	0	0	0

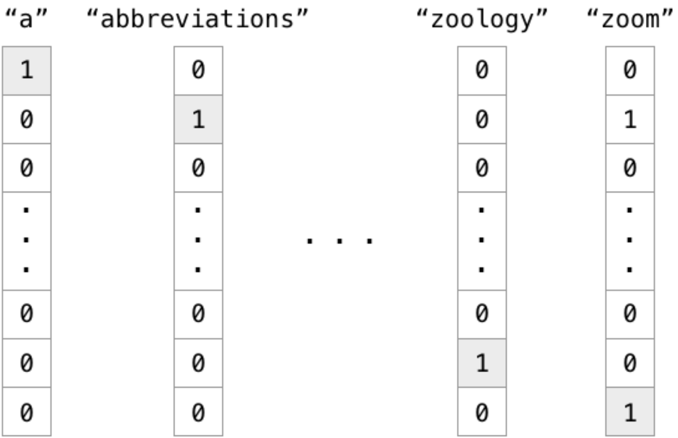
Recurrent Neural Networks - Motivation

x:	$x^{<1>}$	$x^{<2>}$	...	$x^{<t>}$	...	$x^{<T_x>}$
x:	Kate	Farrahi	...	Hare	...	learning
y:	$y^{<1>}$	$y^{<2>}$	...	$y^{<t>}$	...	$y^{<T_y>}$
y:	1	1	...	1	...	0

In this example, $T_x = T_y = 8$ but T_x and T_y can be different.

One Hot Encoding

How can we represent individual words?



[1]

---

[1] https://ayearofai.com

Why Not a Standard Feed Forward Network?

- For a task such as "Named Entity Recognition" a feed forward network would have several disadvantages

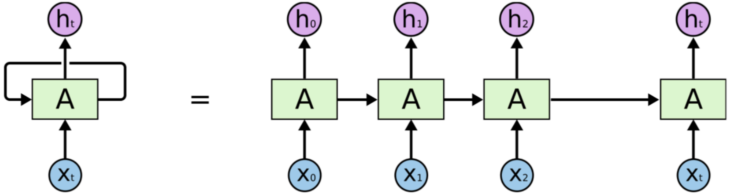
Why Not a Standard Feed Forward Network?

- For a task such as "Named Entity Recognition" a feed forward network would have several disadvantages
- The inputs and outputs may have varying lengths

Why Not a Standard Feed Forward Network?

- For a task such as "Named Entity Recognition" a feed forward network would have several disadvantages
- The inputs and outputs may have varying lengths
- The features wouldn't be shared across different positions in the network

Recurrent Neural Networks

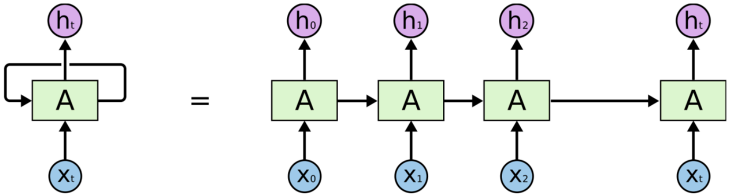


[2]

- RNNs are a family of ANNs for processing sequential data

[2] Image taken from `https://towardsdatascience.com`

Recurrent Neural Networks

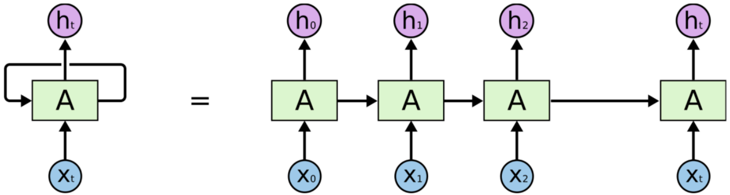


[2]

- RNNs are a family of ANNs for processing sequential data
- RNNs have directed cycles in their computational graphs

[2] Image taken from `https://towardsdatascience.com`

Recurrent Neural Networks

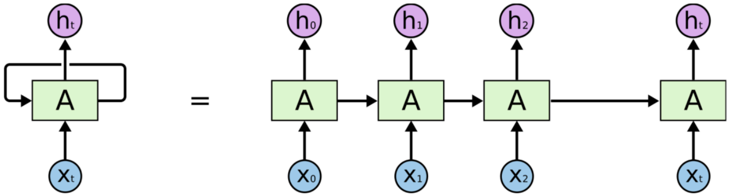


[2]

- RNNs are a family of ANNs for processing sequential data
- RNNs have directed cycles in their computational graphs
- They can have complicated dynamics, difficult to train

[2] Image taken from `https://towardsdatascience.com`

Recurrent Neural Networks



[2]

- RNNs are a family of ANNs for processing sequential data
- RNNs have directed cycles in their computational graphs
- They can have complicated dynamics, difficult to train
- They are more biologically realistic

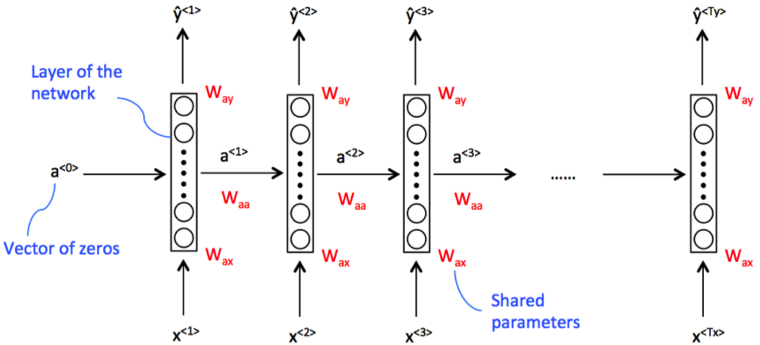
[2] Image taken from `https://towardsdatascience.com`

Recurrent Neural Networks

RNNs combine two properties which make them very powerful.

1. Distributed hidden state that allows them to store a lot of information about the past efficiently. This is because several different units can be active at once, allowing them to remember several things at once.
2. Non-linear dynamics that allows them to update their hidden state in complicated ways. They can however have complicated dynamics, making them difficult to train

Backpropagation Through Time (BPTT)



[3]

[3] Image taken from Andrew Ng

BPTT - Forward Pass

$$a^{<t>} = g(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a) \tag{1}$$

BPTT - Forward Pass

$$a^{<t>} = g(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a) \quad (1)$$

$$\hat{y}^{<t>} = g(w_{ya}a^{<t>} + b_y) \quad (2)$$

BPTT - Forward Pass

$$\begin{aligned}
a^{<t>} &= g(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a) && (1)\\
\hat{y}^{<t>} &= g(w_{ya}a^{<t>} + b_y) && (2)\\
\mathcal{L}^{<t>} &= -y^{<t>}log(\hat{y}^{<t>}) - (1 - y^{<t>})log(1 - \hat{y}^{<t>}) && (3)
\end{aligned}$$

BPTT - Forward Pass

$$
\begin{aligned}
a^{<t>} &= g(w_{aa}a^{<t-1>} + w_{ax}x^{<t>} + b_a) & (1)\\
\hat{y}^{<t>} &= g(w_{ya}a^{<t>} + b_y) & (2)\\
\mathcal{L}^{<t>} &= -y^{<t>}log(\hat{y}^{<t>}) - (1 - y^{<t>})log(1 - \hat{y}^{<t>}) & (3)\\
\mathcal{L} &= \sum_{t=1}^{T_y} \mathcal{L}^{<t>} & (4)
\end{aligned}
$$

BPTT - Backwards Pass

$$\frac{\partial \mathcal{L}^{<3>}}{\partial w_{ya}} = \frac{\partial \mathcal{L}^{<3>}}{\partial \hat{y}^{<3>}} \frac{\partial \hat{y}^{<3>}}{\partial w_{ya}} \tag{5}$$

BPTT - Backwards Pass

$$\frac{\partial \mathcal{L}^{<3>}}{\partial w_{ya}} = \frac{\partial \mathcal{L}^{<3>}}{\partial \hat{y}^{<3>}} \frac{\partial \hat{y}^{<3>}}{\partial w_{ya}} \tag{5}$$

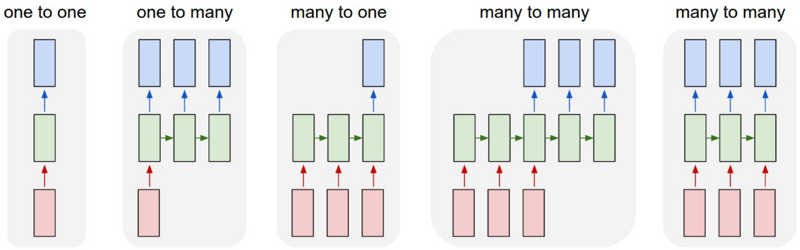
$$\frac{\partial \mathcal{L}^{<3>}}{\partial w_{aa}} = \frac{\partial \mathcal{L}^{<3>}}{\partial \hat{y}^{<3>}} \frac{\partial \hat{y}^{<3>}}{\partial a^{<3>}} \frac{\partial \hat{a}^{<3>}}{\partial w_{aa}} \tag{6}$$

$$\tag{7}$$

BPTT - Backwards Pass

$$
\begin{aligned}
\frac{\partial \mathcal{L}^{<3>}}{\partial w_{ya}} &= \frac{\partial \mathcal{L}^{<3>}}{\partial \hat{y}^{<3>}} \frac{\partial \hat{y}^{<3>}}{\partial w_{ya}} && (5) \\
\frac{\partial \mathcal{L}^{<3>}}{\partial w_{aa}} &= \frac{\partial \mathcal{L}^{<3>}}{\partial \hat{y}^{<3>}} \frac{\partial \hat{y}^{<3>}}{\partial a^{<3>}} \frac{\partial \hat{a}^{<3>}}{\partial w_{aa}} && (6) \\
& && (7) \\
\text{Recall } a^{<3>} &= g(w_{aa} a^{<2>} + w_{ax} x^{<3>} + b_a) \\
\frac{\partial \mathcal{L}^{<3>}}{\partial w_{aa}} &= \frac{\partial \mathcal{L}^{<3>}}{\partial \hat{y}^{<3>}} \frac{\partial \hat{y}^{<3>}}{\partial a^{<3>}} \frac{\partial a^{<3>}}{\partial a^{<2>}} \frac{\partial a^{<2>}}{\partial a^{<1>}} \frac{\partial a^{<1>}}{\partial w_{aa}} && (8)
\end{aligned}
$$

Recurrent Neural Networks



[4]

[4] http://karpathy.github.io/2015/05/21/rnn-effectiveness/