RNNs

The problem: Want to model sequences of arbitrary length.

- Stock prices
- Physiological data
- Genetic code

...

Suppose we have $x_1, x_2, \ldots, x_{t-1}$ and want to predict x_t.

$$X_t \sim P(x_t \mid \underbrace{x_1, x_2, \ldots, x_{t-1}}_{\text{size of this depends on } t!})$$

So far we have handled via Markov assumption

$$P(x_t \mid x_1 \ldots x_{t-1}) = P(x_t \mid x_{t-1})$$

$$\underbrace{P(x_1, x_2, \ldots, x_T) = P(x_1)\, P(x_2 \mid x_1)\, P(x_3 \mid x_2) \ldots P(x_T \mid x_{T-1})}_{}$$

First-order Markov

Pros?

- Terms will (likely) be well estimated

- Very simple

Cons?:

- Obviously wrong

Enter recurrent neural networks (RNNs). These pack history into a fixed size vector h_t s.t.

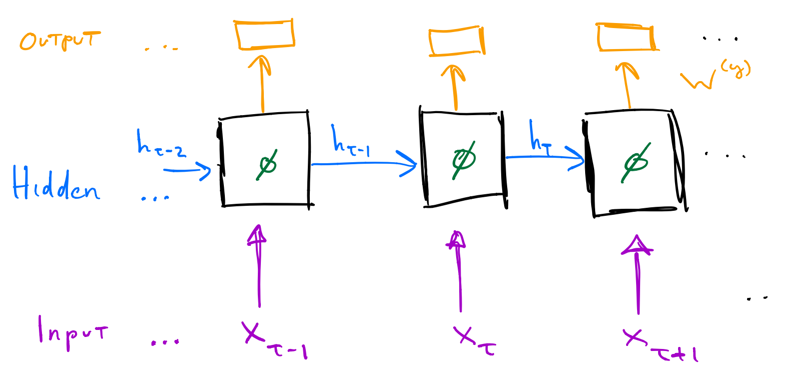
$$P(x_t \mid x_1, x_2, \ldots, x_{t-1}) = P(x_t \mid h_t)$$

Context vector, updated at each time step (word)

$$\underset{(1\times h)}{h_t} = \overset{\text{Activation}}{\phi}\Big\{ \underbrace{x_t}_{\substack{\text{Word at}\\ \text{position } t}}\!\!\underset{(1\times d)\,(d\times h)}{W^{(x)}} + \underbrace{h_{t-1}}_{\substack{\text{last hidden}\\ \text{state}}}\!\!\underset{(1\times h)\,(h\times h)}{W^{(h)}} + \underset{(1\times h)}{b^{(h)}} \Big\} \quad \text{Model parameters}$$

(Figure below)

- The model is recurrent in that it consumes its own output from the previous step
- Consequently, it can be passed over inputs of arbitrary length: the number of params is fixed!



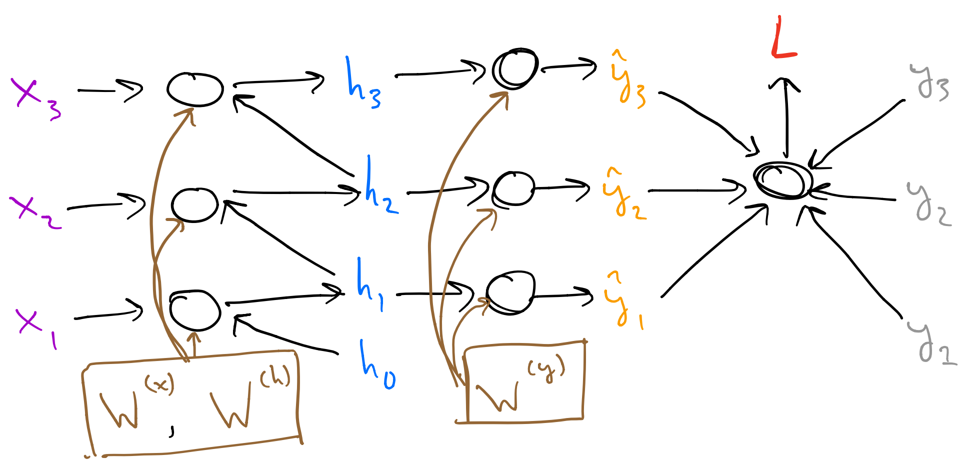
<u>Downside</u> Requires sequentially processing input!

Can we avoid this?

The output layer will be some function of h. In language modeling this might just be a dense + softMax over the vocab.

Let's think about backprop in RNNs.

$$L = \frac{1}{T} \sum_{t=1}^{T} \mathcal{L}(\hat{y}_t, y_t)$$



$$\frac{\partial L}{\partial \hat{y}_t} = \frac{1}{T} \cdot \frac{\partial \ell(\hat{y}_t, y_t)}{\partial \hat{y}_t}$$

$$\frac{\partial L}{\partial W^{(y)}} = \sum_{\tau=1}^{T} \frac{\partial L}{\partial \hat{y}_\tau} \cdot \frac{\partial \hat{y}_\tau}{\partial W^{(y)}}$$

The hidden vectors a bit trickier due to recurrence. Consider the last state h_3

$$\frac{\partial L}{\partial h_3} = \frac{\partial L}{\partial \hat{y}_3} \cdot \frac{\partial \hat{y}_3}{\partial h_3}$$

How about h_2? Gradient flows in from $\hat{y}_2$ and from $\hat{y}_3$ via h_3: So we add these

$$\frac{\partial L}{\partial h_2} = \frac{\partial L}{\partial \hat{y}_2} \cdot \frac{\partial \hat{y}_2}{\partial h_2} + \frac{\partial L}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2}$$

Finally, similar for h_1

$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial \hat{y}_1} \cdot \frac{\partial \hat{y}_1}{\partial h_1} + \frac{\partial L}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1}$$

These are used to derive gradients for $W^{(h)}$ and $W^{(x)}$:

$$\frac{\partial L}{\partial W^{(h)}} = \sum_{\tau=1}^{T} \frac{\partial L}{\partial h_\tau} \cdot \frac{\partial h_\tau}{\partial W^{(h)}}$$

As sequences grow, gradients must travel "backward" further and further. Consider

$$\left\| \frac{\partial h_t}{\partial h_k} \right\| = \left\| \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right\|$$

Jacobians!

Assume upper-bound $\left\| \frac{\partial h_j}{\partial h_{j-1}} \right\| \leq \beta$

$$\rightarrow \leq \beta^{t-k}$$

So if $t-k$ is big we can see how this might be problematic!

Specifically: vanishing when the error signal attenuates. Alternatively, gradients may explode for the same reason.

Jane walked into the room. John walked in too. It was late in the day. John said hi to ____.

[ex. from Socher, cs224d]

Next time, we'll intro RNN variants that try to address this.