Neural Language Models

- Very basic neural LM : $P(w_i \mid w_{i-1}) = \frac{e^{V_i * C_{i-1}}}{\sum_{w' \in V} e^{V_w * C_{i-1}}}$

- More generally : $P(w_i \mid w_1 \dots w_{i-1}) = \text{softmax}(U_{w_i} \cdot f(w_1 \dots w_{i-1}))$

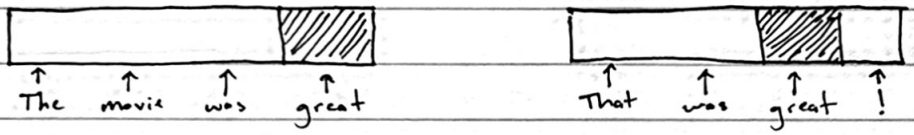
f = neural net to embed the context

- f is Deep Averaging Network? → Ignores Order!
- f is Fed Forward Neural Net? → Doesn't scale to long contexts

* How can recurrent neural nets solve issue of scalability?

RNN's and their Shortcomings

* Feed-forward Neural Nets can't handle variable length input
  - Each position in feature vector has fixed semantics

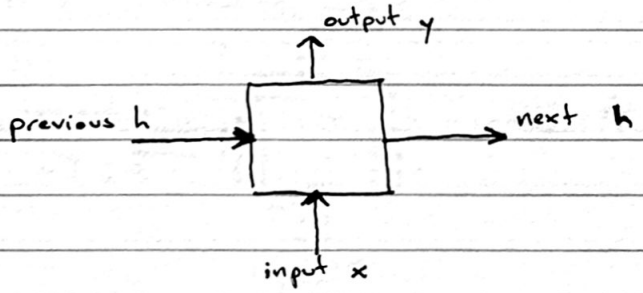


- These don't look related (great is in two different orthogonal subspaces)

* Instead, we need to:
  1. Process each word in uniform way
  2. ... while still exploiting context that token occurs in

• **RNN Abstraction**

RNN defined in terms of a cell that takes some input x, has some hidden state h, and updates that hidden state and produces output y (all vector-valued)



* **Issues: Vanishing Gradient**

- Gradient diminishes going through tanh; if not in $[-2, 2]$, gradient ≈ 0
- Repeated multiplication by V causes problems

* Slow. Don't parallelize and there are $O(n)$ non-parallel operations to encode n items