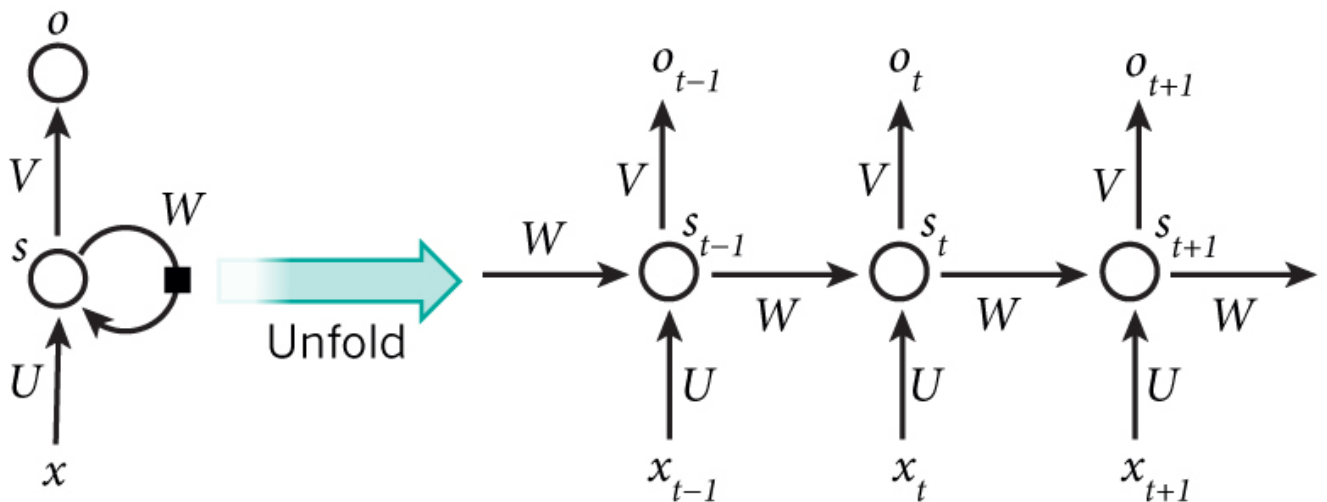


# Recurrent Neural Network



## 1. Model

input:

$$x = (x_1, x_2, \dots, x_T) \quad x_t \in \mathbb{R}^n$$

initialize hidden state:

$$s_0 \in \mathbb{R}^k$$

forward propagation:

$$s_t = \tanh(Ux_t + Ws_{t-1}) \quad (t = 1, 2, \dots, T)$$

$$\hat{y}_t = \text{softmax}(Vs_t) \quad (t = 1, 2, \dots, T)$$

output:

$$\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T) \quad \hat{y}_t \in \mathbb{R}^m$$

## 2. Backpropagation Through Time

cost function:

$$E(\hat{y}) = \sum_{t=1}^T E_t(\hat{y}_t)$$

definition:

$$\begin{aligned} h_t &= Ux_t + Ws_{t-1} \quad (t = 1, 2, \dots, T) \\ z_t &= Vs_t \quad (t = 1, 2, \dots, T) \end{aligned}$$

gradient for  $V$ :

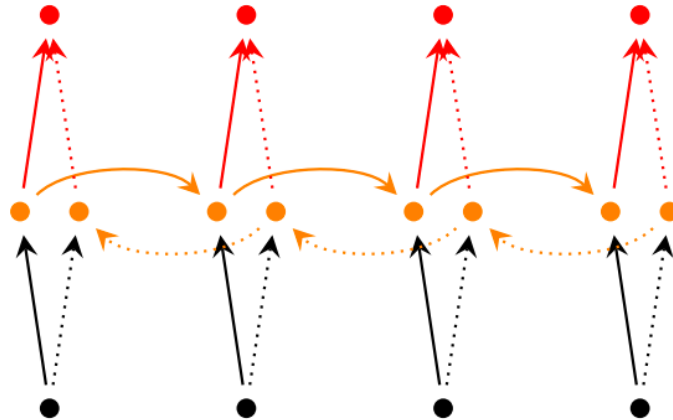
$$\begin{aligned} \frac{\partial E_t}{\partial V} &= \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial V} = \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial V} \\ &= \left( \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t} \right) \cdot s_t^T \quad (\text{need } \hat{y}_t, s_t; t = 1, 2, \dots, T) \end{aligned}$$

gradient for  $W$ :

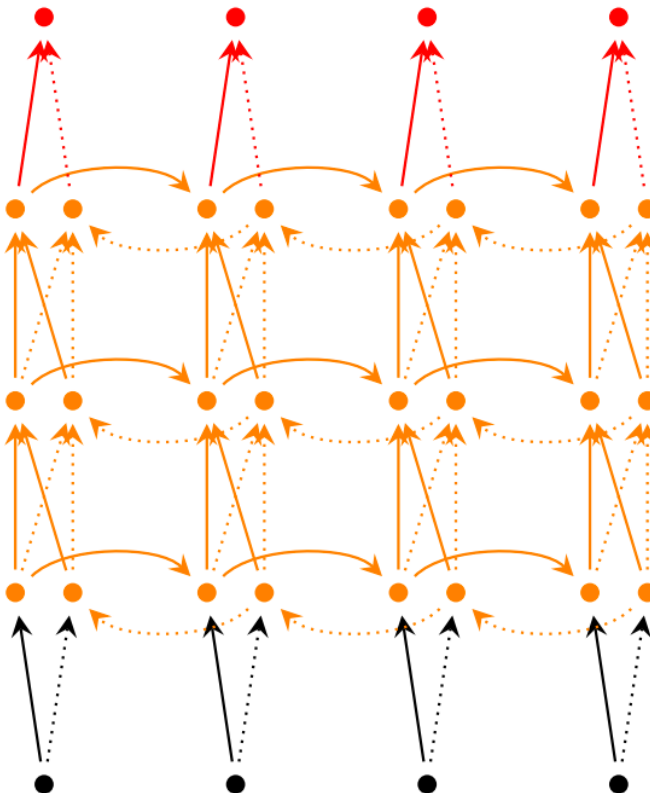
$$\begin{aligned} \frac{\partial s_1}{\partial W} &= \frac{\partial s_1}{\partial h_1} \cdot \frac{\partial h_1}{\partial W} \quad (\text{need } s_1, s_0) \\ \frac{\partial s_t}{\partial W} &= \frac{\partial s_t}{\partial h_t} \cdot \left( \frac{\partial h_t}{\partial W} + W \cdot \frac{\partial s_{t-1}}{\partial W} \right) \quad (\text{need } s_t, s_{t-1}; t = 2, 3, \dots, T) \\ \frac{\partial E_t}{\partial W} &= \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial s_t} \cdot \frac{\partial s_t}{\partial W} \\ &= \left( \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t} \right)^T \cdot V \cdot \frac{\partial s_t}{\partial W} \quad (\text{need } \hat{y}_t; t = 1, 2, \dots, T) \end{aligned}$$

### 3. RNN Extensions

Bidirectional RNNs:

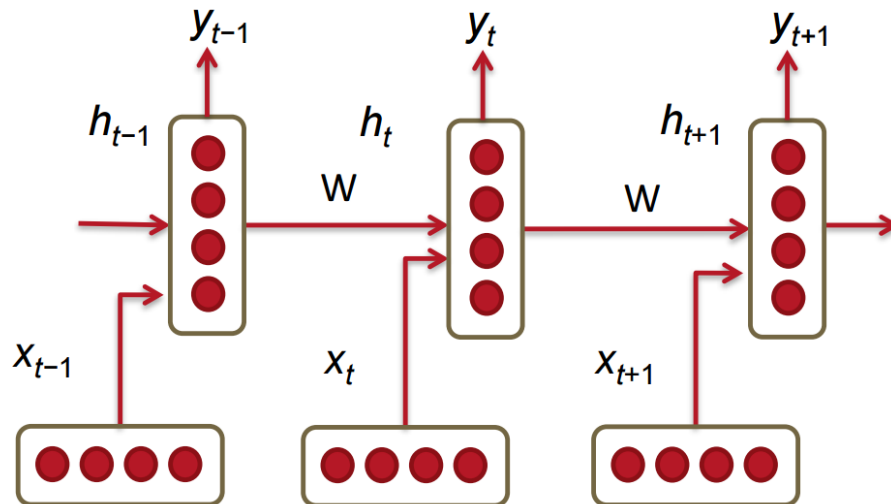


Deep (Bidirectional) RNNs:



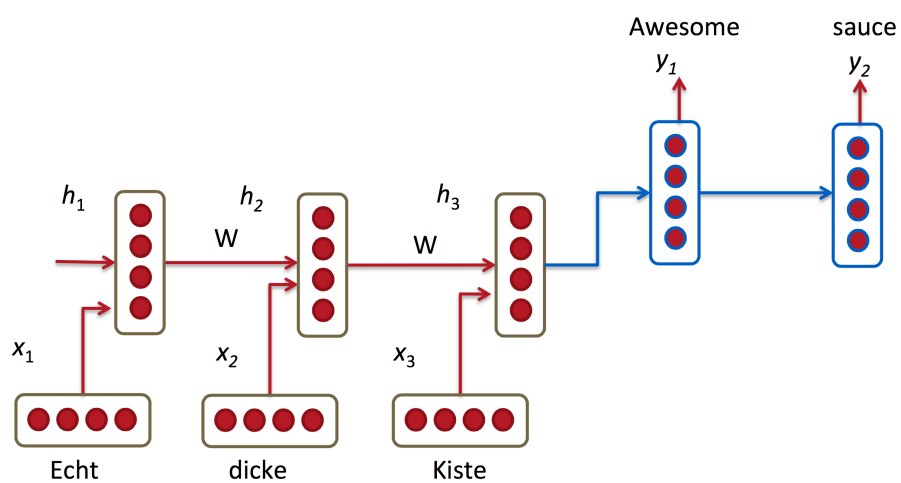
## 4. Applications

### 1. Language Model [1, 2, 3]



*Recurrent neural network based language model*

### 2. Machine Translation [4]



*RNN for Machine Translation*

# Reference

1. Mikolov, Tomas, et al. "Recurrent neural network based language model." INTERSPEECH. Vol. 2. 2010.  
[http://www.fit.vutbr.cz/research/groups/speech/servite/2010/rnnlm\\_mikolov.pdf](http://www.fit.vutbr.cz/research/groups/speech/servite/2010/rnnlm_mikolov.pdf)
2. Recurrent Neural Network Language Models: <http://www.rnnlm.org/>
3. Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks: <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
4. Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014. <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
5. A. Graves. Supervised Sequence Labelling with Recurrent Neural Networks. Textbook, Studies in Computational Intelligence, Springer, 2012.  
<https://www.cs.toronto.edu/~graves/preprint.pdf>