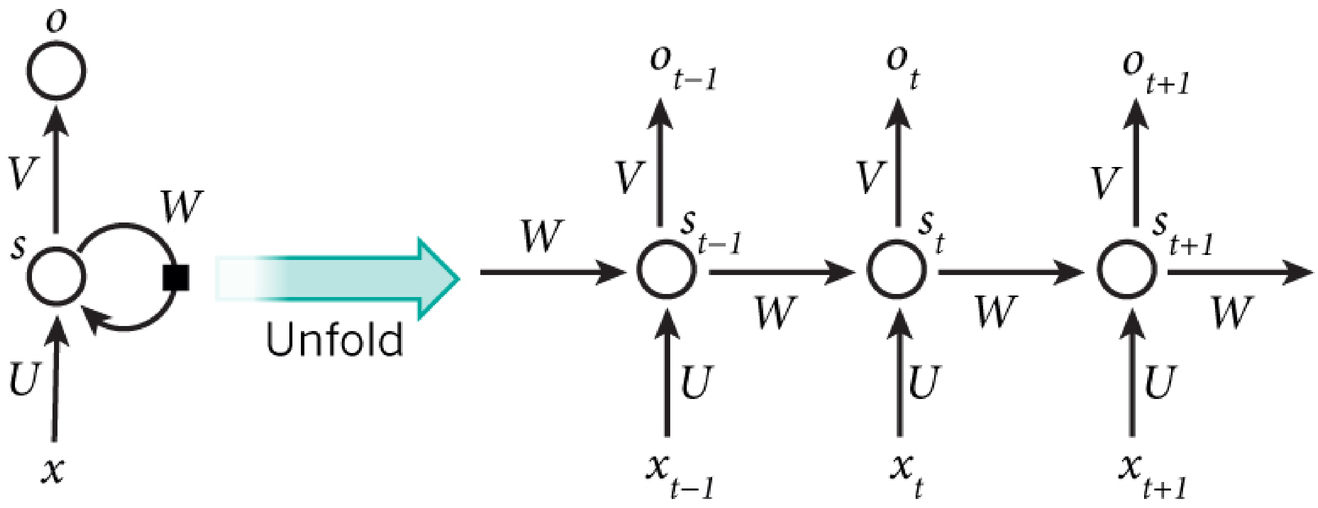
Machine Learning

Recurrent Neural Network



1. Basics

sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma'(x) = \sigma(x) \cdot [1 - \sigma(x)]$$

hyperbolic function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\tanh'(x) = 1 - \tanh^2(x)$$

rectified linear unit(ReLU):

$$f(x) = \max(0, x)$$

softmax function:

$$\mathbf{y} = \text{softmax}(\mathbf{x})$$

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

$$\frac{\partial y_i}{\partial x_j} = \begin{cases} -y_i \cdot y_j, & i \neq j \\ y_i \cdot (1 - y_i), & i = j \end{cases}$$

2. Model

input:

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

initialize hidden state:

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

forward propagation:

$$\begin{aligned} s_t &= \tanh(Ux_t + Ws_{t-1}) \quad (t = 1, 2, \ldots, T) \\ \hat{y}_t &= \text{softmax}(Vs_t) \quad (t = 1, 2, \ldots, T) \end{aligned}$$

output:

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

3. Backpropagation Through Time

cost function:

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

definition:

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

gradient for V:

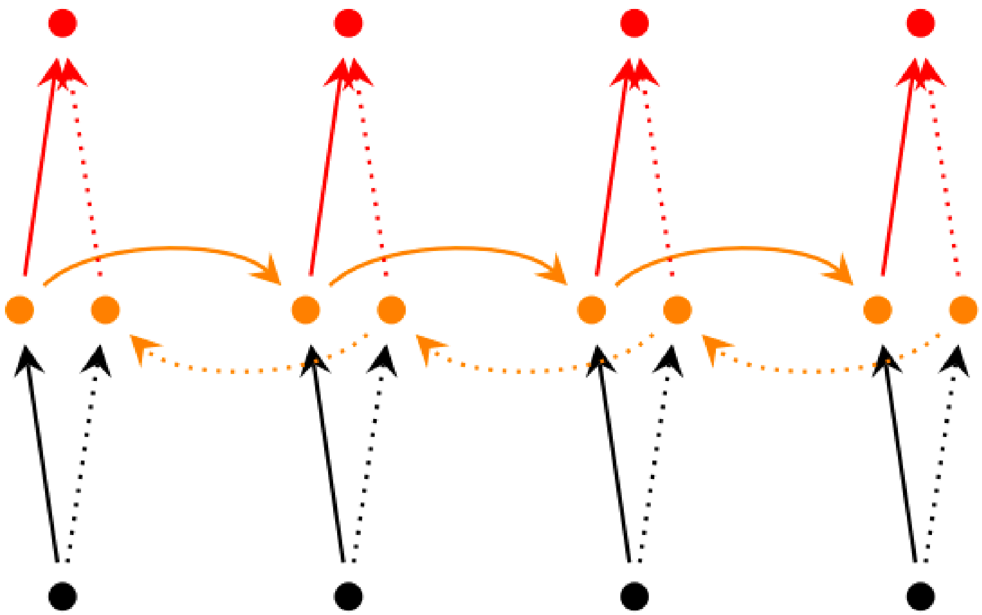
$$\frac{\partial E_t}{\partial V} = \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial V} = \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t} \cdot \frac{\partial z_t}{\partial V}$$
$$= \left(\frac{\partial E_t}{\partial \hat{y}_t} \cdot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

gradient for W:

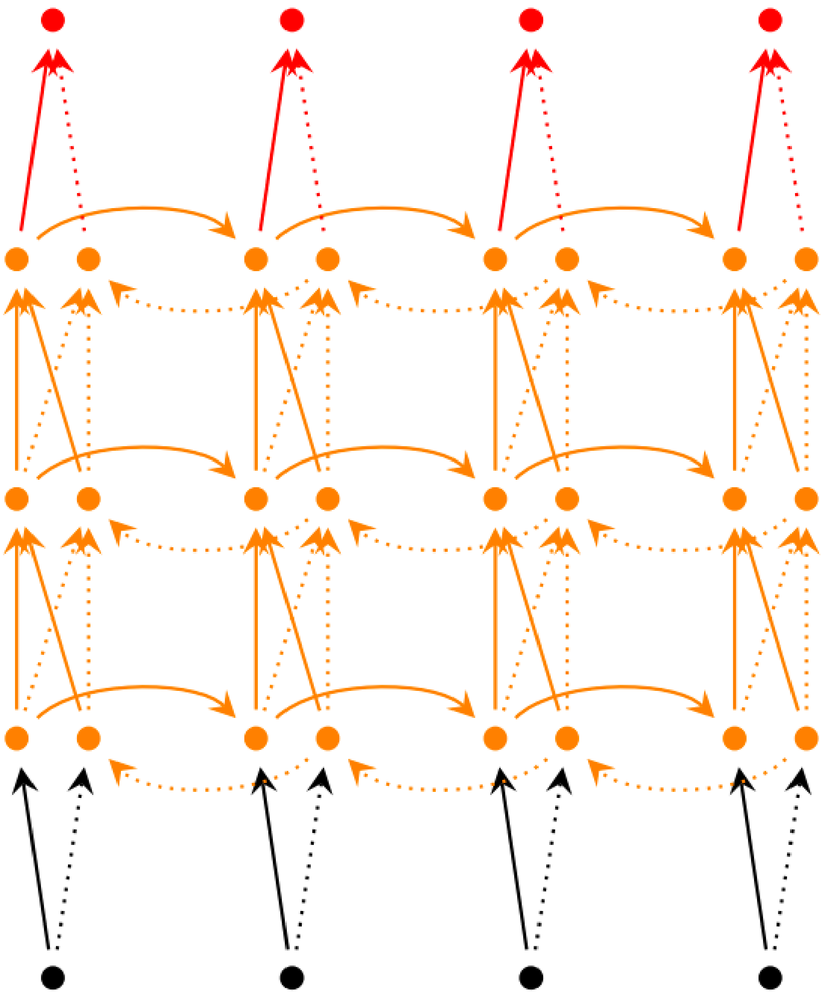
$$\frac{\partial s_1}{\partial W} = \frac{\partial s_1}{\partial h_1} \cdot \frac{\partial h_1}{\partial W} \quad (\text{need } s_1, s_0)$$
$$\frac{\partial s_t}{\partial W} = \frac{\partial s_t}{\partial h_t} \cdot \left(\frac{\partial h_t}{\partial W} + W \cdot \frac{\partial s_{t-1}}{\partial W} \right)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

4. RNN Extensions

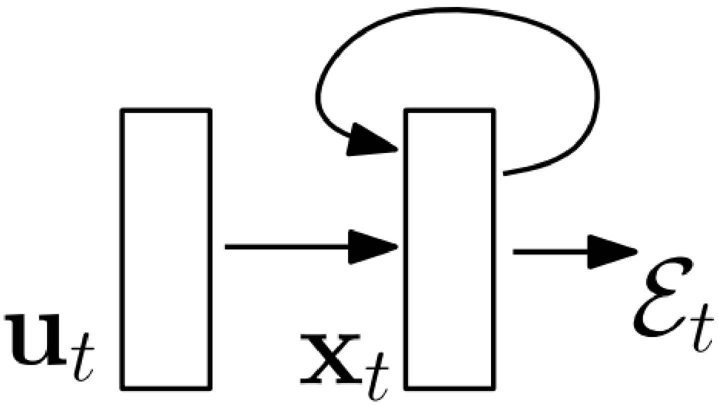
Bidirectional RNNs:



Deep (Bidirectional) RNNs:



5. Vanishing Gradient in RNN [1]



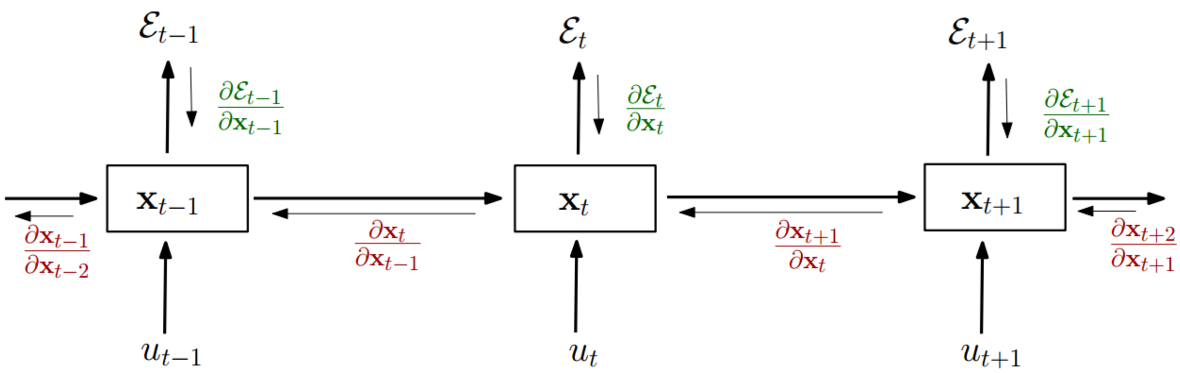
hidden state:

$$\mathbf{x}_t = \mathbf{W}_{rec}\sigma(\mathbf{x_{t-1}}) + \mathbf{W}_{in}\mathbf{u}_t + \mathbf{b}$$

cost:

$$\mathcal{E} = \sum_{1 \leq t \leq T} \mathcal{E}_t = \sum_{1 \leq t \leq T} \mathcal{L}(\mathbf{x}_t)$$

unrolling RNN:



gradients:

$$\frac{\partial \mathcal{E}}{\partial \theta} = \sum_{1 \leq t \leq T} \frac{\partial \mathcal{E}_t}{\partial \theta}$$

$$\frac{\partial \mathcal{E}_t}{\partial \theta} = \sum_{1 \leq k \leq t} \left(\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t} \frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} \frac{\partial^+ \mathbf{x}_k}{\partial \theta} \right)$$

$$\frac{\partial \mathbf{x}_t}{\partial \mathbf{x}_k} = \prod_{t \geq i > k} \frac{\partial \mathbf{x}_i}{\partial \mathbf{x}_{i-1}} = \prod_{t \geq i > k} \mathbf{W}_{rec}^T \mathrm{diag}(\sigma'(\mathbf{x}_{i-1}))$$

proof:

it is sufficient for $\lambda_1 < \frac{1}{\gamma}$, where λ_1 is the largest singular value of $\mathbf{W}_{rec}$ and $\left\|\mathrm{diag}(\sigma'(\mathbf{x}_k))\right\| \leq \gamma \in \mathcal{R}$, for the vanishing gradient problem to occur.

$$\forall k, \left\| \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_k} \right\| \leq \left\|\mathbf{W}_{rec}^T\right\| \left\|\mathrm{diag}(\sigma'(\mathbf{x}_k))\right\| < \frac{1}{\gamma}\gamma < 1$$

let $\eta \in \mathcal{R}$ be such that $\forall k, \left\| \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_k} \right\| \leq \eta < 1$.

$$\left\|\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t}\left(\prod_{i=k}^{t-1}\frac{\partial \mathbf{x}_{i+1}}{\partial \mathbf{x}_i}\right)\right\| \leq \eta^{t-k}\left\|\frac{\partial \mathcal{E}_t}{\partial \mathbf{x}_t}\right\|$$

deal with the exploding and vanishing gradient:

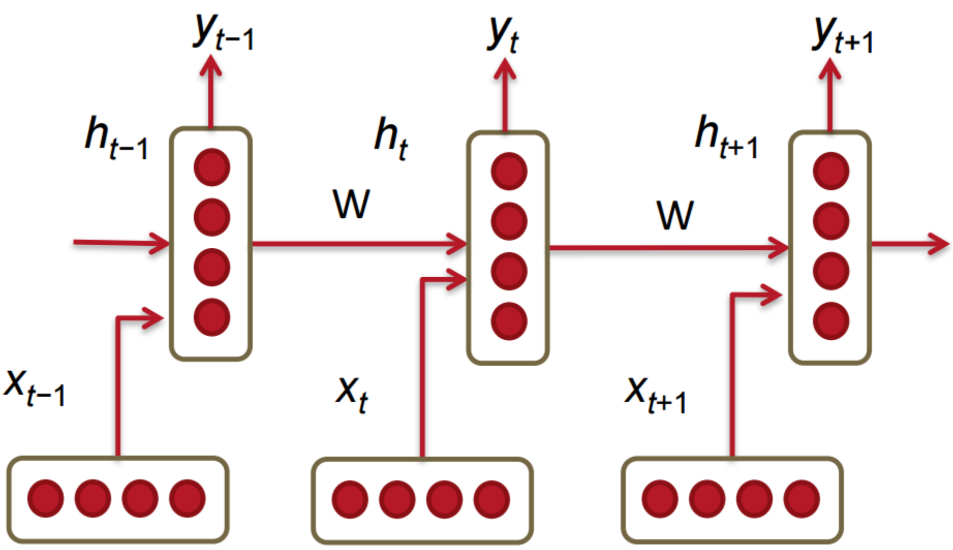
- $L1$ or $L2$ penalty
- LSTM
- clipping gradient

gradient flow in LSTM:

$$\frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_k} = \frac{\partial \mathbf{c}_t}{\partial \mathbf{c}_{t-1}} \cdots \frac{\partial \mathbf{c}_{k+1}}{\partial \mathbf{c}_k} = \mathrm{diag}(\mathbf{f}_t)\cdots\mathrm{diag}(\mathbf{f}_k) = \mathrm{diag}(\mathbf{f}_t \odot \cdots \odot \mathbf{f}_k)$$

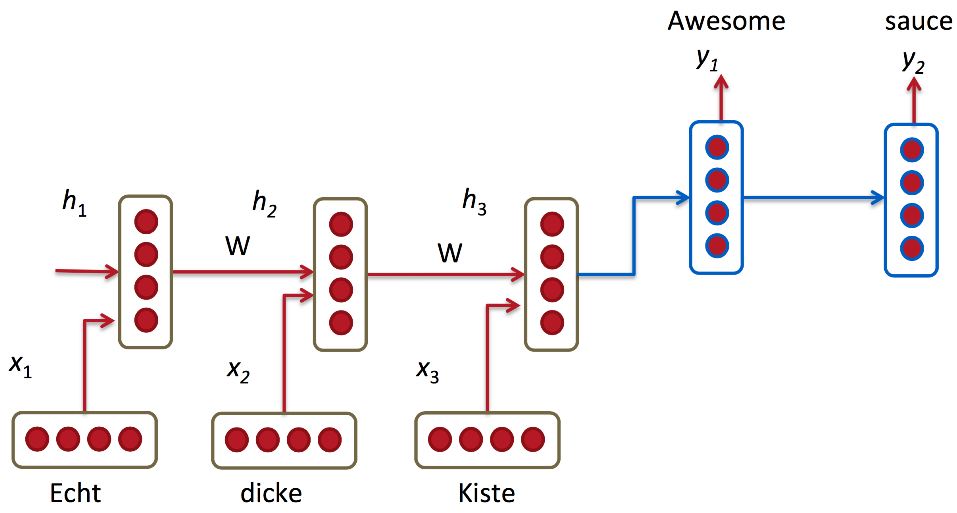
6. Applications

Language Model [2, 3, 4]:



Recurrent neural network based language model

Machine Translation [5]:



RNN for Machine Translation

Reference

1. Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." Proceedings of The 30th International Conference on Machine Learning. 2013. http://www.jmlr.org/proceedings/papers/v28/pascanu13.pdf
2. 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
3. Recurrent Neural Network Language Models: http://www.rnnlm.org/
4. Andrej Karpathy. The Unreasonable Effectiveness of Recurrent Neural Networks: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
5. 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
6. A. Graves. Supervised Sequence Labelling with Recurrent Neural Networks. Textbook, Studies in Computational Intelligence, Springer, 2012. https://www.cs.toronto.edu/~graves/preprint.pdf