Bài 12: Recurrent Neural Network

VietAI Teaching Team

Nội dung

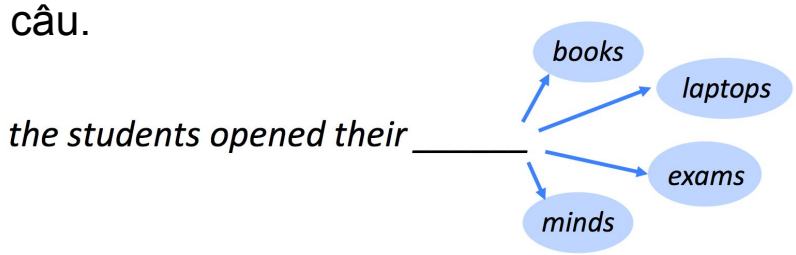
- Language Models (1 bài toán trong NLP)

  ↓ Motivates

- Giới thiệu RNN
  - Ví dụ thực tiễn
  - Kiến trúc
- Vấn đề và các hướng giải quyết
  - Vanishing và Exploding Gradients

1 Language Model

- Language Model là bài toán dự đoán từ xuất hiện tiếp theo trong câu.



- Cho trước *(m-1)* từ, LM tính xác suất để tìm ra từ m tiếp theo.

$$P(w_m|w_1, ..., w_{m-1})$$

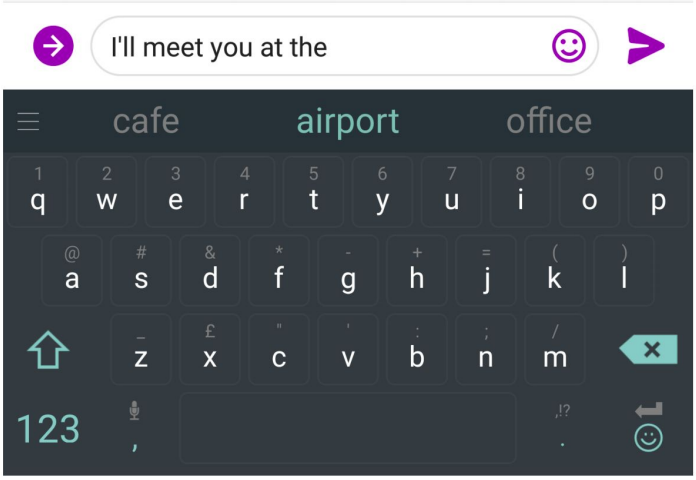
1 Language Model

- Language Model tính xác suất của một chuỗi các từ (kí tự):

$$P(w_1, ..., w_m) = \prod_{i=1}^{m} P(w_i|w_1, ..., w_{i-1})$$

- Sử dụng trong nhiều ứng dụng:
  - Sắp xếp các từ đúng thứ tự
  - Sử dụng từ đúng ngữ pháp (grammatical/syntactic)
  - Sử dụng từ đúng ngữ nghĩa (semantics)

1 Language Model --- Application



1 Language Model --- Application

Google

what is the

what is the **weather**
what is the **meaning of life**
what is the **dark web**
what is the **xfl**
what is the **doomsday clock**
what is the **weather today**
what is the **keto diet**
what is the **american dream**
what is the **speed of light**
what is the **bill of rights**

Google Search I'm Feeling Lucky

1 Language Model

- Sử dụng xác suất có điều kiện và giả định Markov (Markov assumption):

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

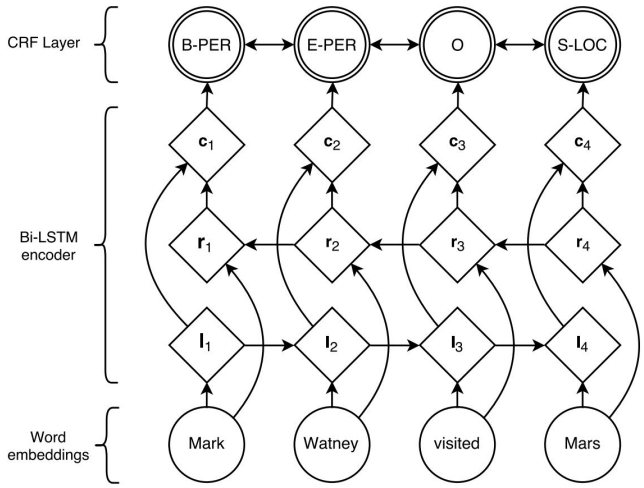
- Ước lượng các xác suất trên dựa vào phương pháp thống kê (đếm) trên tập dữ liệu văn bản cho trước:

$$p(w_2|w_1) = \frac{\text{count}(w_1, w_2)}{\text{count}(w_1)}$$

$$p(w_3|w_1, w_2) = \frac{\text{count}(w_1, w_2, w_3)}{\text{count}(w_1, w_2)}$$

2 RNN

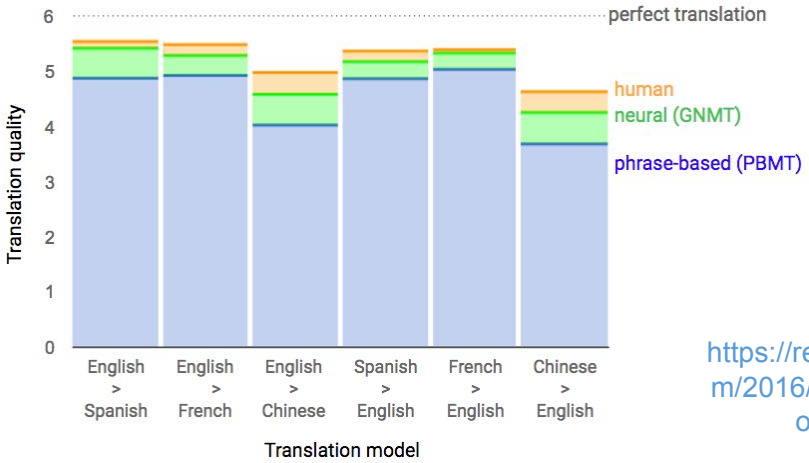
Ví dụ thực tiễn - Named Entity Recognition



Lample et al., Neural Architectures for Named Entity Recognition, NAACL 2016.

2 RNN

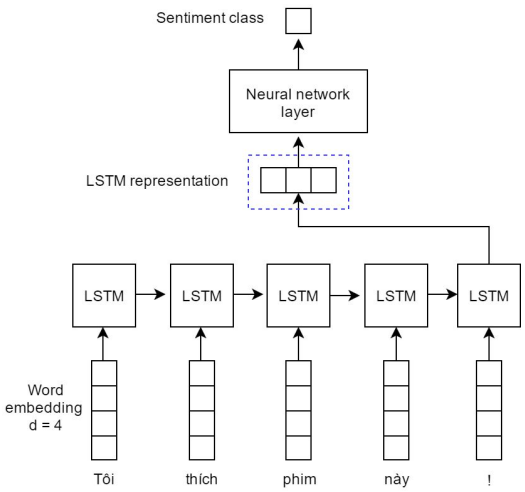
Ví dụ thực tiễn - Machine Translation



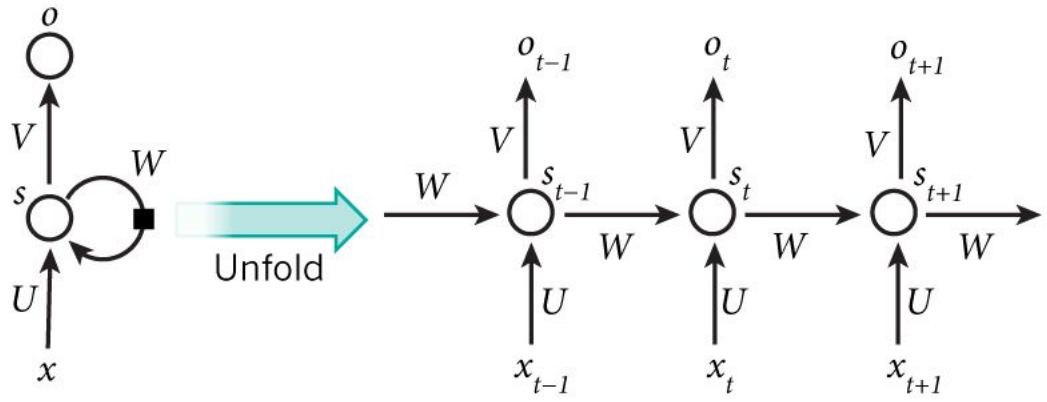
https://research.googleblog.com/2016/09/a-neural-network-for-machine.html

2 RNN

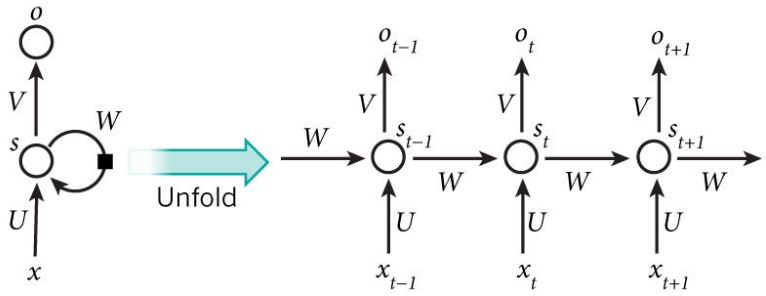
Ví dụ thực tiễn - Sentiment Analysis



2 RNN
Kiến trúc



2 RNN
Kiến trúc



- Input: *x*
- Hidden state: *h* hoặc *s*
- Output: *o* hoặc $\hat{y}$
- Tham số mô hình: *U, V, W*

2 RNN
Kiến trúc

Given list of word **vectors**: $x_1, \ldots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T$

At a single time step: $s_t = \sigma(Ux_t + Ws_{t-1})$

$\hat{y}_t = softmax(Vs_t)$

$\hat{P}(x_{t+1} = v_j | x_t, \ldots, x_1) = \hat{y}_{t,j}$

2 RNN
Kiến trúc

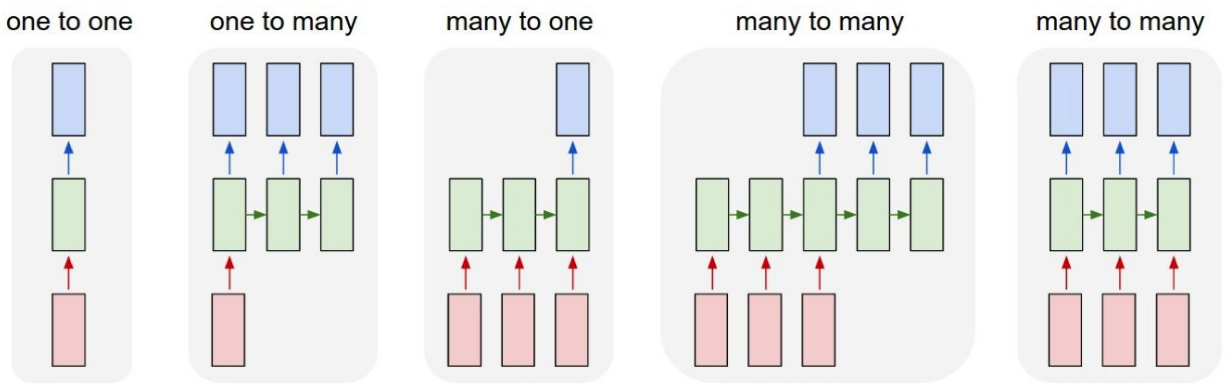
- Tại thời điểm *t*, cost function có dạng như sau:

$$J^{(t)}(\theta) = -\sum_{j=1}^{|V|} y_{t,j} \log \hat{y}_{t,j}$$

- Trong cả khoảng thời gian *T*, ta có:

$$J = -\frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{|V|} y_{t,j} \log \hat{y}_{t,j}$$

2 RNN
Kiến trúc

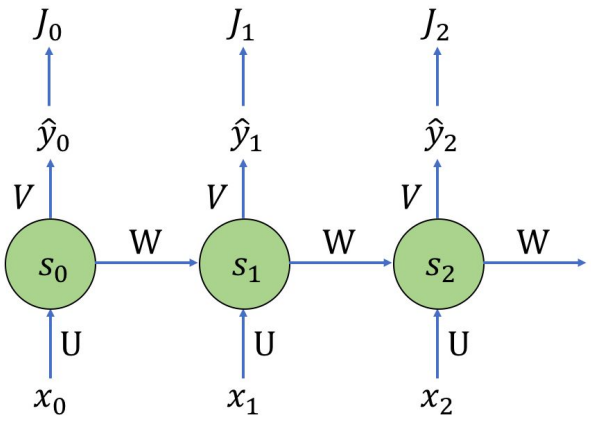


3 Vấn đề chính
Long term dependencies and short term attention

"In France, I had a great time and I learnt some of the ____ language."

our parameters are not trained to capture long-term dependencies, so the word we predict will mostly depend on the previous few words, not much earlier ones

3 Vấn đề Vanishing/Exploding Gradient
Nắm vấn đề



loss at time $t = J_t(\Theta)$

Θ = our parameters, like weights

total loss = $J(\Theta) = \Sigma_t J_t(\Theta)$

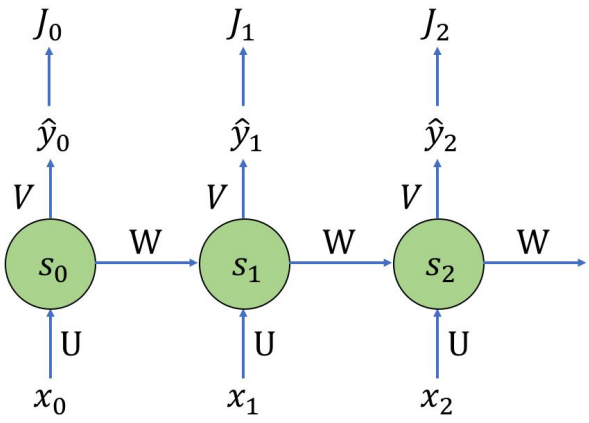
3 Vấn đề Vanishing/Exploding Gradient
Nắm vấn đề

Đạo hàm của cost function theo ma trận trọng số

$$\frac{\partial J}{\partial V} = \sum_t \frac{\partial J_t}{\partial V} \qquad \frac{\partial J}{\partial W} = \sum_t \frac{\partial J_t}{\partial W}$$

3 Vấn đề Vanishing/Exploding Gradient

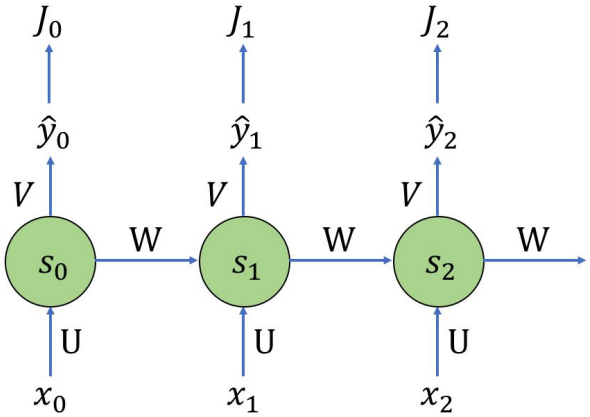
Nắm vấn đề



$$\frac{\partial J_2}{\partial W} = \frac{\partial J_2}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial s_2}\frac{\partial s_2}{\partial W}$$

3 Vấn đề Vanishing/Exploding Gradient

Nắm vấn đề



$$\frac{\partial J_2}{\partial W} = \frac{\partial J_2}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial s_2}\frac{\partial s_2}{\partial W}$$

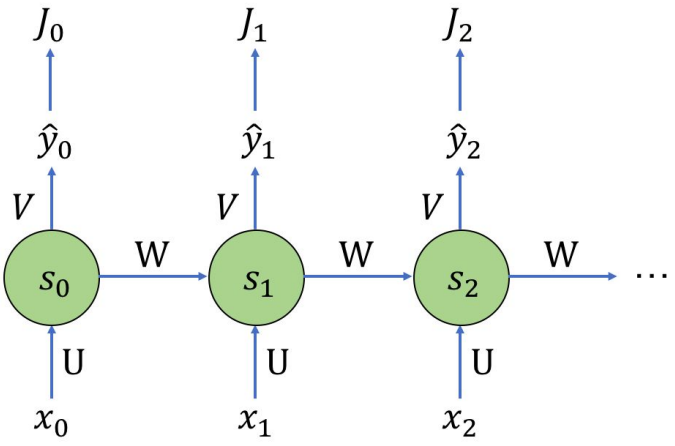
But wait...

$$s_2 = \sigma(Ux_2 + Ws_1)$$

s_1 also depends on W so we can't just treat $\frac{\partial s_2}{\partial W}$ as a constant!

3 Vấn đề Vanishing/Exploding Gradient

Nắm vấn đề



$$\frac{\partial s_2}{\partial W} + \frac{\partial s_2}{\partial s_1}\frac{\partial s_1}{\partial W} + \frac{\partial s_2}{\partial s_0}\frac{\partial s_0}{\partial W}$$

3 Vấn đề Vanishing/Exploding Gradient

Nắm vấn đề

$$\frac{\partial J_2}{\partial W} = \sum_{k=0}^{2} \frac{\partial J_2}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial s_2}\frac{\partial s_2}{\partial s_k}\frac{\partial s_k}{\partial W}$$

Contributions of *W* in previous timesteps to the error at timestep *t*

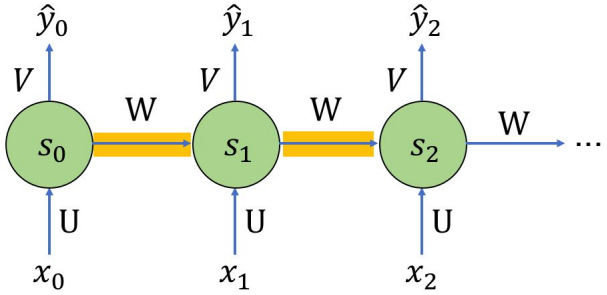
3 Vấn đề Vanishing/Exploding Gradient

Nắm vấn đề

$$\frac{\partial J_2}{\partial W} = \sum_{k=0}^{2} \frac{\partial J_2}{\partial \hat{y}_2} \frac{\partial \hat{y}_2}{\partial s_2} \frac{\partial s_2}{\partial s_k} \frac{\partial s_k}{\partial W}$$

Tại k = 0

$$\frac{\partial s_2}{\partial s_0} = \frac{\partial s_2}{\partial s_1} \frac{\partial s_1}{\partial s_0}$$



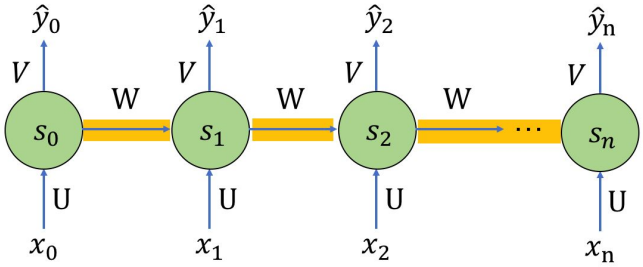
3 Vấn đề Vanishing/Exploding Gradient

Nắm vấn đề

$$\frac{\partial J_n}{\partial W} = \sum_{k=0}^{n} \frac{\partial J_n}{\partial \hat{y}_n} \frac{\partial \hat{y}_n}{\partial s_n} \frac{\partial s_n}{\partial s_k} \frac{\partial s_k}{\partial W}$$

Tại k = 0

$$\frac{\partial s_n}{\partial s_{n-1}} \frac{\partial s_{n-1}}{\partial s_{n-2}} \cdots \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial s_1} \frac{\partial s_1}{\partial s_0}$$



3 Vấn đề Vanishing/Exploding Gradient
Nắm vấn đề

what are each of these terms?

$$\frac{\partial s_n}{\partial s_{n-1}} = W^T diag[f'(W_{s_{j-1}} + Ux_j)]$$

$$\frac{\partial s_n}{\partial s_{n-1}} \frac{\partial s_{n-1}}{\partial s_{n-2}} \cdots \frac{\partial s_3}{\partial s_2} \frac{\partial s_2}{\partial s_1} \frac{\partial s_1}{\partial s_0}$$

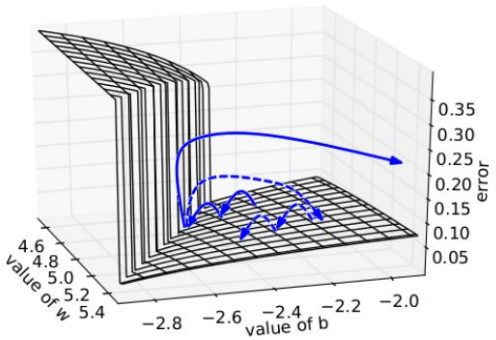
W = sampled from standard normal distribution = mostly < 1

f = tanh or sigmoid so **f'** < 1

3 Exploding Gradient
Hướng khắc phục: Gradient Clipping

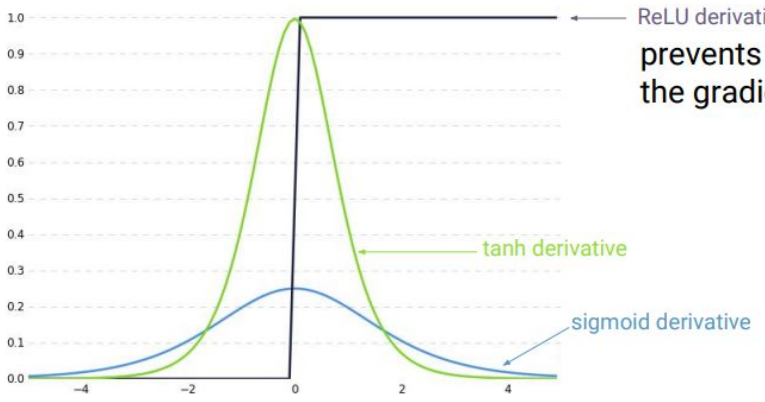
Algorithm 1 Pseudo-code for norm clipping the gradients whenever they explode

$\hat{\mathbf{g}} \leftarrow \frac{\partial \mathcal{E}}{\partial \theta}$
if $\|\hat{\mathbf{g}}\| \geq threshold$ **then**
  $\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|} \hat{\mathbf{g}}$
end if



3 Khắc phục

Phương án 1: Thay đổi hàm kích hoạt



3 Khắc phục

Phương án 2: Khởi tạo lại ma trận trọng số

weights initialized to identity matrix ⟶ $I_n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$

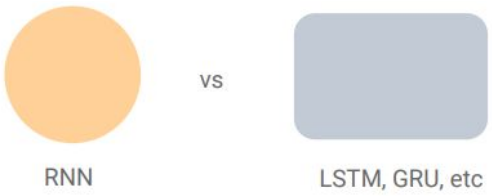
biases initialized to zeros

prevents W from shrinking the gradients

3 Khắc phục

Phương án 3: Thay đổi lại node của RNN

Thay vì dùng RNN cell đơn giản, có thể dùng các cơ chế cổng (gates) để tính toán hidden states từ đó có thể kiểm soát được thông tin trong RNN



Tài liệu tham khảo

1. CS224n, Stanford University
2. Deep Learning Book, Ian Goodfellow, Yoshua Bengio and Aaron Courville
3. http://www.wildml.com/2015/10/recurrent-neural-networks-tutorial-part-3-backpropagation-through-time-and-vanishing-gradients/
4. Learning a long term dependencies is difficult http://www.dsi.unifi.it/~paolo/ps/tnn-94-gradient.pdf
5. Deep Learning Summer School at Montreal 2016 and 2017