연 세 대 학 교 통 계 학 회 ESC

20FALL Week3

LM and RNN, Facny RNN 추가자료

ADSTOREPOST.COM

09. 2020

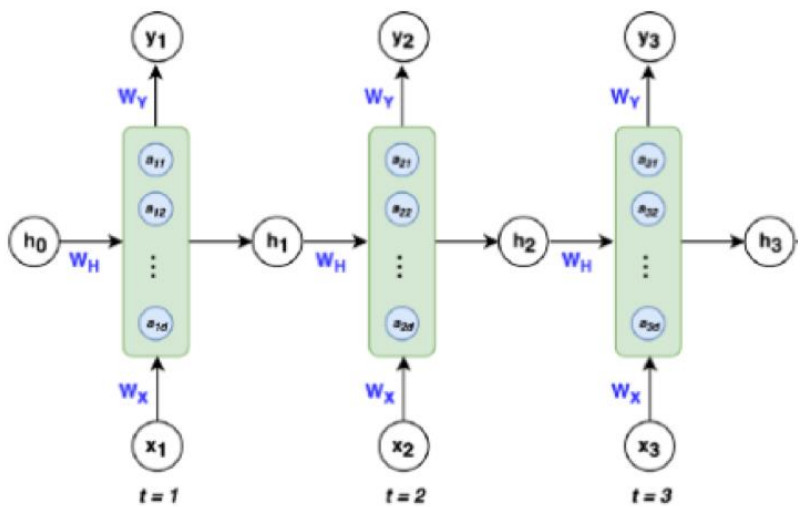
How it does work?

A toy example
"dogs"

How it does work?

A toy example

"dogs"

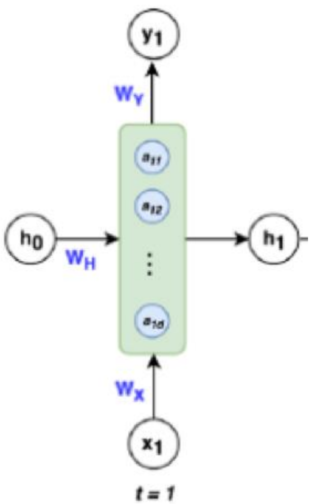


How it does work?

A toy example

"dogs"

$$\begin{bmatrix}0.1 & 0.5 & 0.1\\0.5 & 0.9 & 0.3\\0.3 & 0.2 & 0.1\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}0.6 & 0.8 & 0.4 & 0.8\\0.2 & 0.2 & 0.8 & 0.7\\0.9 & 0.8 & 0.1 & 0.2\end{bmatrix}\begin{bmatrix}1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}0.6\\0.2\\0.9\end{bmatrix}$$



$$h_1 = sigmoid\begin{bmatrix}0.6\\0.2\\0.9\end{bmatrix}=\begin{bmatrix}0.65\\0.55\\0.71\end{bmatrix}$$

$$\hat{y}_1 = softmax\left(\begin{bmatrix}0.9 & 0.8 & 0.3\\0.2 & 0.3 & 0.4\\0.6 & 0.9 & 0.1\\0.5 & 0.0 & 0.3\end{bmatrix}\begin{bmatrix}0.65\\0.55\\0.71\end{bmatrix}\right)=\begin{bmatrix}0.36\\0.19\\0.27\\0.18\end{bmatrix}$$

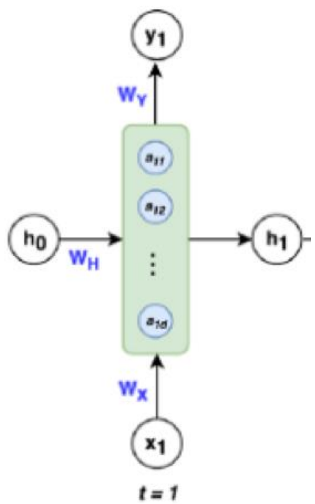
How does it work?

A toy example
"dogs"

$$W_X^+ = W_X - \gamma \frac{\partial L}{\partial W_X}$$

$$W_Y^+ = W_Y - \gamma \frac{\partial L}{\partial W_Y}$$

$$W_H^+ = W_H - \gamma \frac{\partial L}{\partial W_H}$$



$$L(y, \hat{y}) = -y' \log(\hat{y})$$

$$\frac{\partial L_t}{\partial W_X} = \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial W_x} + \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial W_x} + \cdots + \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-n}}\frac{\partial h_{t-n}}{\partial W_x}$$

How does LSTM prevent the vanishing gradient problem?

R
N
N

속도의 차이

$$\frac{\partial L_t}{\partial W_X} = \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial W_x} + \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial W_x} + \cdots + \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t}\frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial h_{t-n}}\frac{\partial h_{t-n}}{\partial W_x}$$

$$h_t = \sigma(wh_{t-1})$$

$$\frac{\partial h_{t+3}}{\partial h_t} = \prod_{k=1}^{3} w\sigma'(wh_{(t+3)-k})$$

'multiplying weight'(1이 아닌경우 exponentially grow or decay)
'the derivative of a sigmoid'

$$\frac{\partial C_{t+3}}{\partial C_t} = \prod_{k=1}^{3} f_{t+k} = \prod_{k=1}^{3} \sigma(forget\ gate\ input)$$

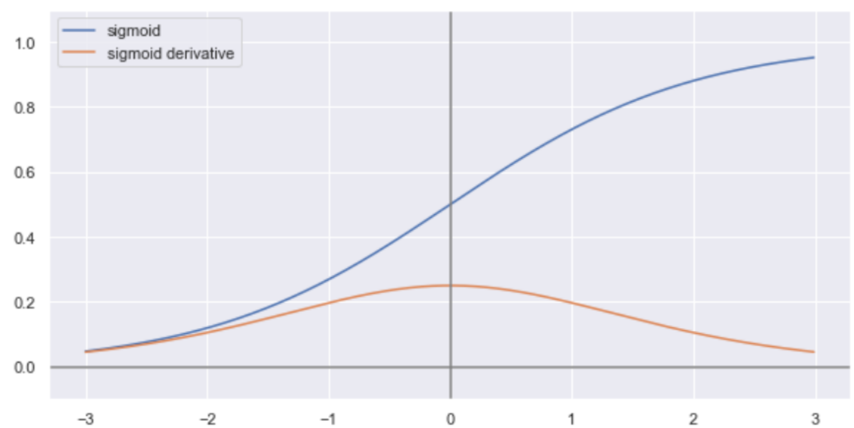
L
S
T
M

Sigmoid vs tanh

tanh를 사용하는 이유

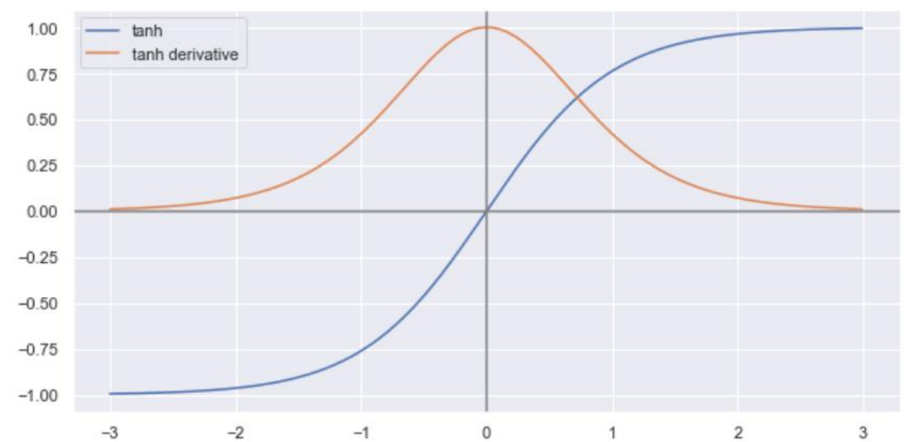
tanh의 미분 최대값이 상대적으로 크기 때문에 gradient 손실을 방지하기 쉽다.

sigmoid



https://towardsdatascience.com/why-data-should-be-normalized-before-training-a-neural-network-c626b7f66c7d

tanh

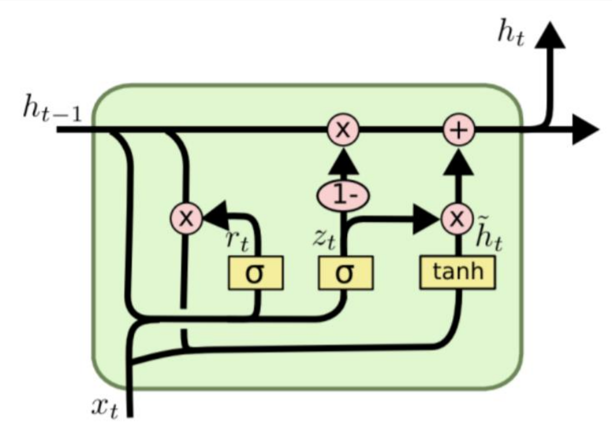


https://towardsdatascience.com/why-data-should-be-normalized-before-training-a-neural-network-c626b7f66c7d

GRU Model

Structure

01



Reset gate(r) (과거 정보에 대한 reset 담당)
Update gate(z)

02

GRU

Forget gate와 input gate를 update gate(z)로 합침.
cell state와 hidden state를 합침.
그 외 몇가지 수정

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$
$$\widetilde{h_t} = \tanh(W_{xh}x_t + W_{hh}(r_t * h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}$$

GRU Model

Structure

01

LSTM

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$
$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
$$\widetilde{C}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

02

GRU

Forget gate와 input gate를 update gate(z)로 합침.
cell state와 hidden state를 합침.
그 외 몇가지 수정

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$
$$\widetilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t * h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t$$

Empirical Evaluation of GRNN on Sequence Modeling

GRU와 LSTM의 공통점 및 차이점

공 통 점

Additivity nature
(Both methods keep the existing content and add the new content on top of it.)

- 인풋이 긴 경우에도 특징을 잘 포착할 수 있다.

- RNN은 replacement의 특성이 강하다. 왜냐하면 계속 값을 덮어씌우기 때문이다.(이전의 히든스테이트와 현재의 새로운 new value로 계속 히든스테이트 변경)

- LSTM에서는 cell state, gru에서는 cell state의 역할을 하는 h가 있기 때문이다.

- Backpropagation을 단순하게 만들어 줘서 vanishing을 방지해준다.

차 이 점

L S T M	G R U
output gate가 존재한다.	output gate가 존재하지 않고 reset gate가 존재한다. 파라미터 수가 LSTM에 비해서 적다.
최종 출력 계산 시 tanh를 적용한다.	출력 계산 시 비선형 activation을 적용하지 않는다.
$C_t = f_t * C_{t-1} + i_t * C_t$ forget과 input의 과정이 독립적으로 작동한다.	$h_t = (1 - z_t) * h_{t-1} + z_t * h_t$ forget gate과 input의 과정이 함께 진행이 된다.
출력값을 한 번 포장한다.	있는 그대로 출력

참조사이트

2020-1학기 Yonsei Univ Applied Statistics 박재우 교수님 딥러닝 강의자료 Recurrent Neural Networks
https://aikorea.org/blog/rnn-tutorial-3/
https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/
https://excelsior-cjh.tistory.com/89
https://excelsior-cjh.tistory.com/185
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://arxiv.org/pdf/1412.3555.pdf
https://brightwon.tistory.com/10
https://curt-park.github.io/2017-04-03/why-is-lstm-strong-on-gradient-vanishing/
개사진: https://www.notepet.co.kr/news/article/article_view/?groupCode=AB130AD130&idx=7277
https://stats.stackexchange.com/questions/185639/how-does-lstm-prevent-the-vanishing-gradient-problem

감사합니다.