Gated Recurrent Unit

허무지

참고 : KyungHyun Cho, Empirical evaluation of Gated Recurrent Neural Networks on Sequence modeling

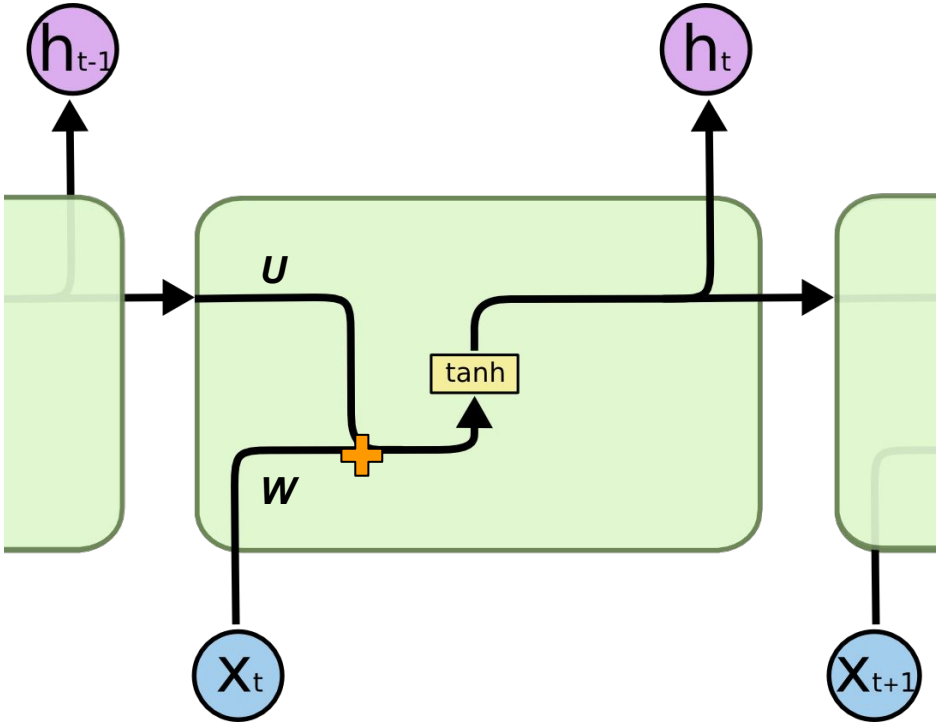
목차

1. Introduction

- RNN 은 입출력의 길이가 가변적이어도 결과가 좋음
- Vanilla RNN보다는 GRU, LSTM의 연구가 많음
- 기계 번역에서는 두 모델 간의 비교가 이미 있음
- 본 논문에서는 polyphonic music datasets로 비교

2. Background

A recurrent hidden state



$$\mathbf{h}_t = \begin{cases} 0, & t = 0 \\ \phi\left(\mathbf{h}_{t-1}, \mathbf{x}_t\right), & \text{otherwise} \end{cases}$$

$$\mathbf{h}_t = g\left(W\mathbf{x}_t + U\mathbf{h}_{t-1}\right),$$

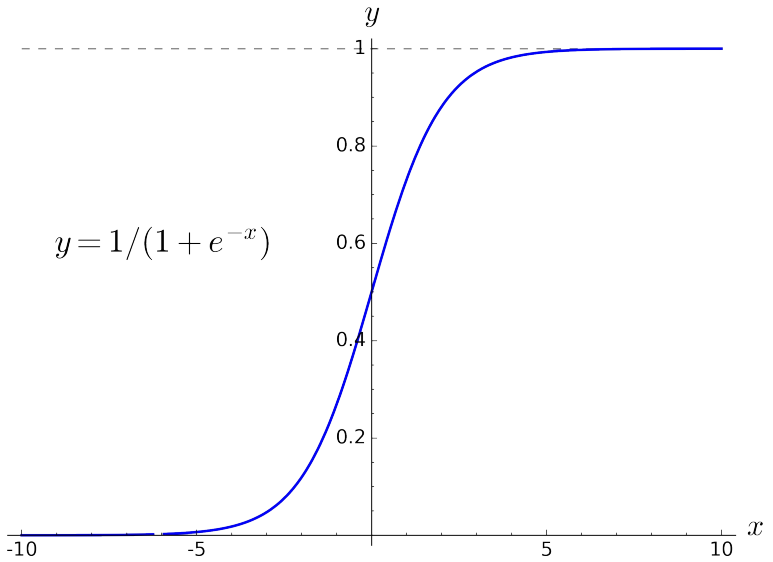
사진 출처 : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

2. Background

Gate

그림 6-13 비유하자면 게이트는 물의 흐름을 제어한다.

그림 6-14 물이 흐르는 양을 0.0~1.0 범위에서 제어한다.

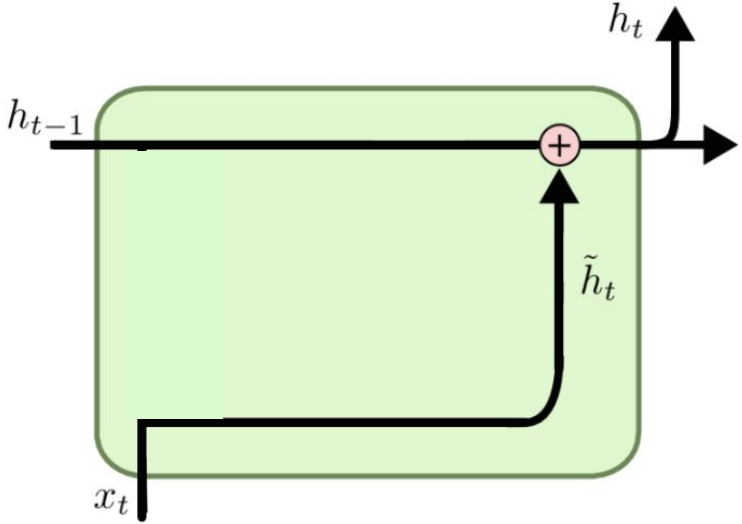


Sigmoid function

사진 출처 :밑바닥으로부터 시작하는 딥러닝 2 , p. 250

3. Simplified GRU

Activation *h*

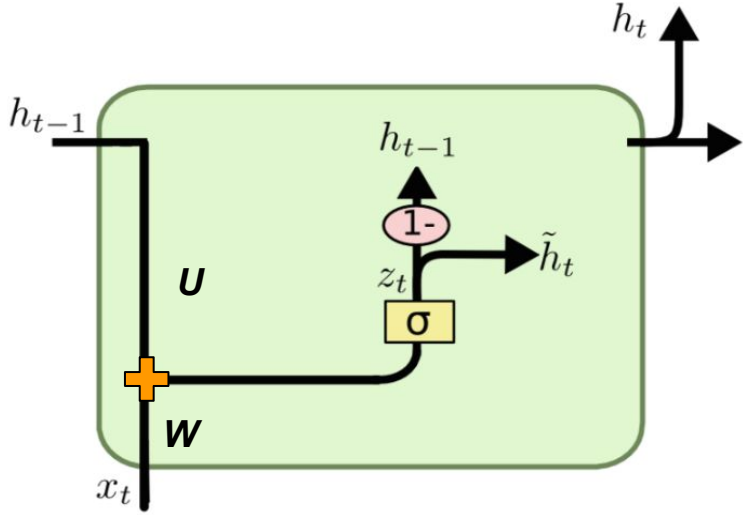


$$h_t^j = \quad h_{t-1}^j + \quad \tilde{h}_t^j$$

사진 출처 : https://feature.engineering/difference-between-lstm-and-gru-for-rnns/

3. Simplified GRU

Update gate z

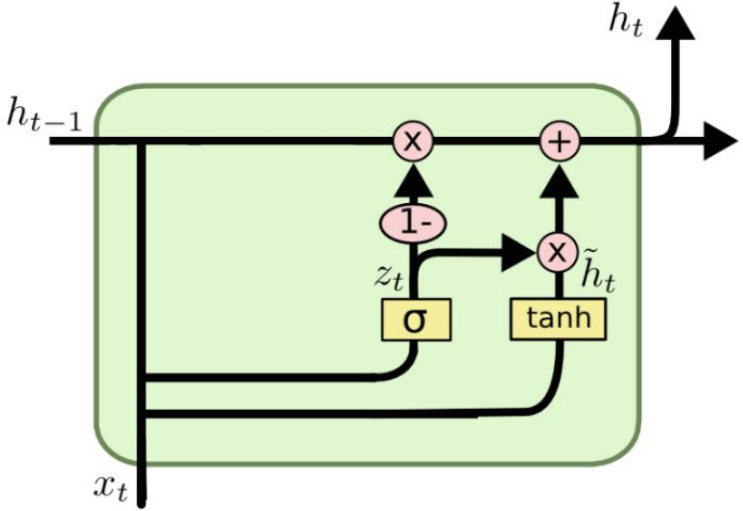


$$z_t^j = \sigma\left(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1}\right)^j$$

사진 출처 : https://feature.engineering/difference-between-lstm-and-gru-for-rnns/

3. Simplified GRU

Update gate z

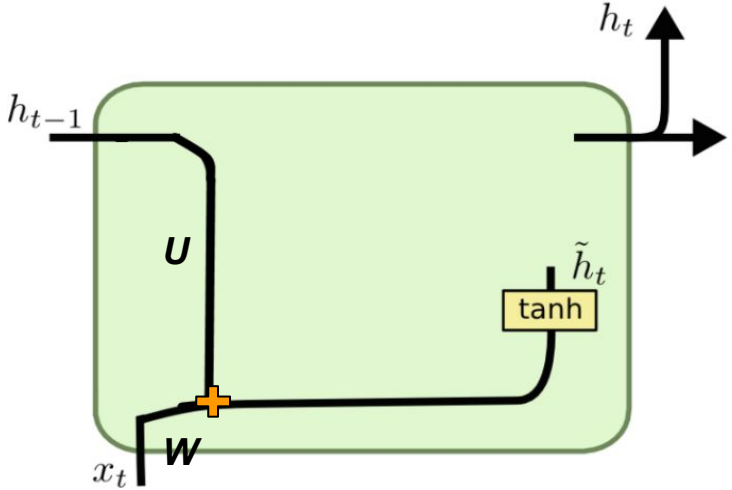


$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j$$

사진 출처 : https://feature.engineering/difference-between-lstm-and-gru-for-rnns/

4. Full GRU

Candidate *h*



$$\tilde{h}_t^j = \tanh\left(W\mathbf{x}_t + U\left(\quad \mathbf{h}_{t-1}\right)\right)^j$$

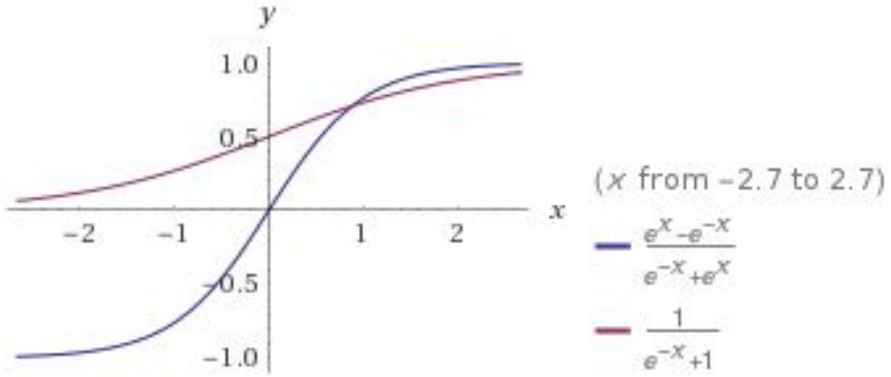
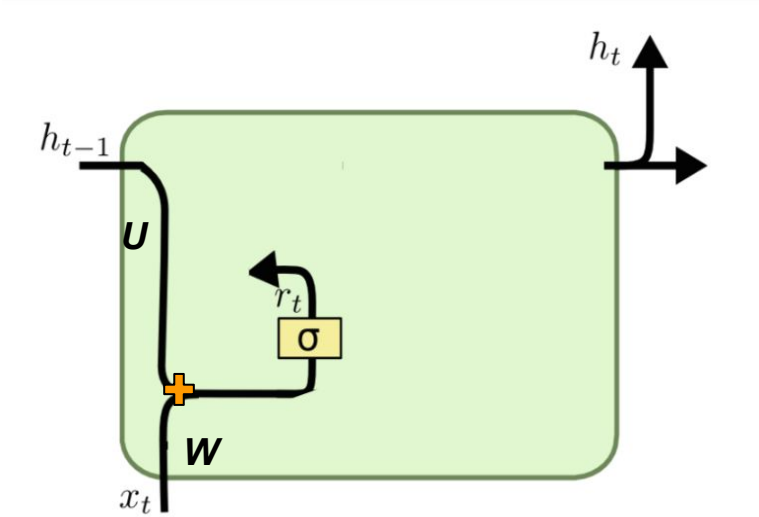


사진 출처 : https://feature.engineering/difference-between-lstm-and-gru-for-rnns/
그래프 출처: https://sebastianraschka.com/faq/docs/tanh-sigmoid-relationship.html

4. Full GRU

Reset gate r

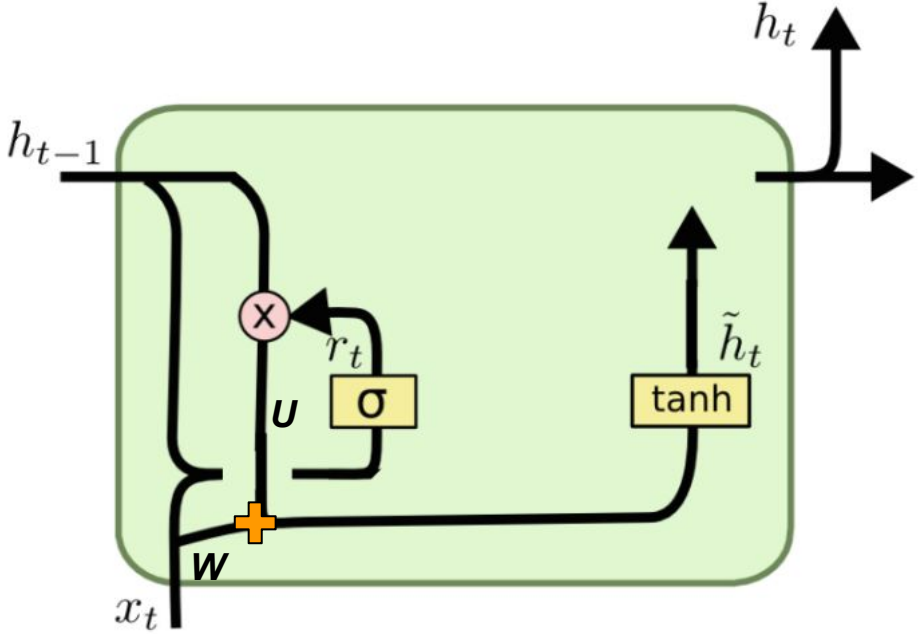


$$r_t^j = \sigma\left(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1}\right)^j$$

사진 출처 : https://feature.engineering/difference-between-lstm-and-gru-for-rnns/

4. Full GRU

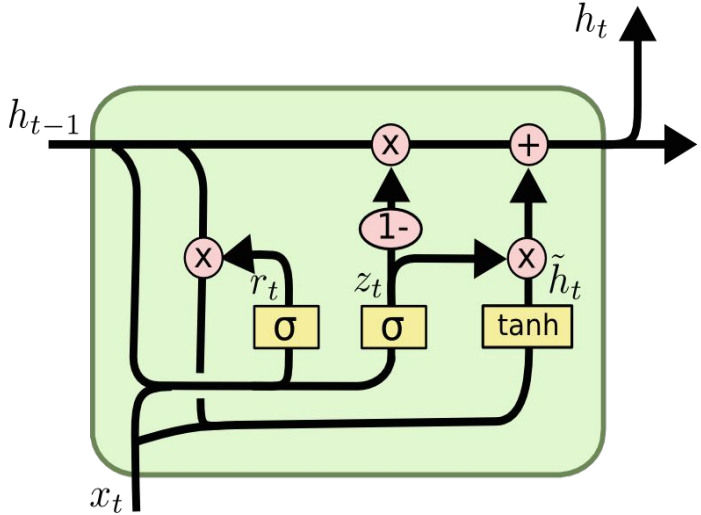
Candidate *h*



$$\tilde{h}_t^j = \tanh\left(W\mathbf{x}_t + U\left(\mathbf{r}_t \odot \mathbf{h}_{t-1}\right)\right)^j$$

사진 출처 : https://feature.engineering/difference-between-lstm-and-gru-for-rnns/

4. Full GRU



$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j \tilde{h}_t^j$$

$$z_t^j = \sigma\left(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1}\right)^j$$

$$r_t^j = \sigma\left(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1}\right)^j$$

$$\tilde{h}_t^j = \tanh\left(W \mathbf{x}_t + U\left(\mathbf{r}_t \odot \mathbf{h}_{t-1}\right)\right)^j$$

5. GRU와 LSTM의 차이

- GRU는 메모리 셀을 따로 가지고 있지 않아

  출력 양을 따로 조정하지 않음.
- LSTM에서는 Input gate와 Forget gate가 독립적이지만

  GRU에서는 Reset gate가 함께 조절 함.

6. Conclusion

- 특정한 데이터 세트에서

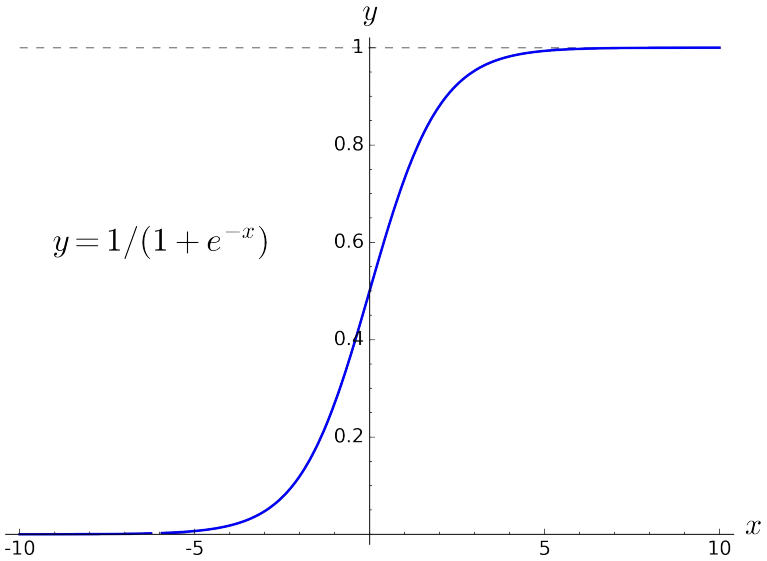
  GRU는 CPU 시간, 파라미터 갱신의 성능이

  LSTM보다 낫다.

- 우위를 논하기에는 연구가 부족하다.

7. 논의점

tanh를 사용하는 이유



Sigmoid function

- 함수 값의 중심이 0이 아님.
  (not zero-centered)
- The range of sigmoid: $0 < \sigma(x) < 1$
- 입력 값(x)이 모두 양수일 경우,
  가중치 w의 기울기는

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial a} * \frac{\partial a}{\partial w}$$

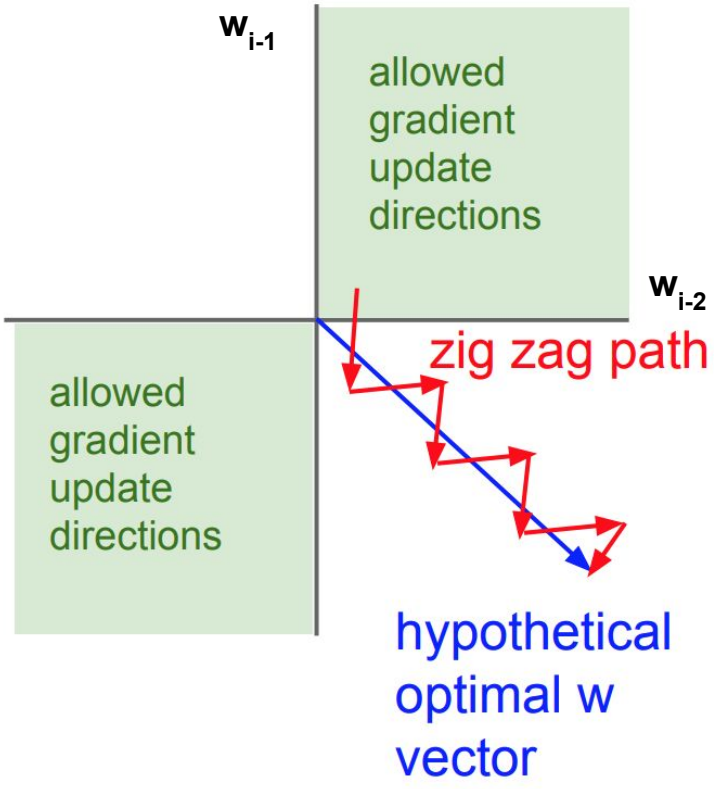
L은 loss 함수, a은 $\mathbf{w}^{\mathbf{T}}\mathbf{x} + b$

- w의 업데이트시 (행렬)
  부호가 일정 함.

출처: http://nmhkahn.github.io/NN

7. 논의점

zigzag path



$\frac{\partial L}{\partial w}$ 업데이트 양(행렬)을 [$w_{1\text{-}1}$, $w_{1\text{-}2}$] 로 가정

- [$w_{1\text{-}1}$, $w_{1\text{-}2}$] 의 부호가 [음수, 음수]일 때 가중치는 음수 방향으로만 움직일 수 있고,
- [$w_{2\text{-}1}$, $w_{2\text{-}2}$] 의 부호가 [양수, 양수]일 때 가중치는 양수 방향으로만 움직일 수 있음.

- 즉, 최적의 w 벡터를 찾기 위해 더 자주 업데이트 해야하는 문제가 발생.

그림 출처 http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture6.pdf
내용 출처 https://nittaku.tistory.com/m/267