Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

허진규 😊

😊 Abstract

- gating machanism을 가진 RNN 위주의 비교를 실시한다.
- polyphonic music modeling과 speech signal modeling을 이용하여 평가했다.
- advanced RNN는 RNN보다 개선되었고, GRU는 LSTM과 비등한 성능이 났다.

😊 Introduction

- LSTM, GRU를 polyphonic music dataset, raw speech data로 실험하겠다.

😊 Backgound : RNN

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

- 해결하기 위한 2가지 접근

1. clipped gradient (hyperparameter 로 threshold 값을 정해서 이를 넘으면 값을 줄인다.)
2. gating unit

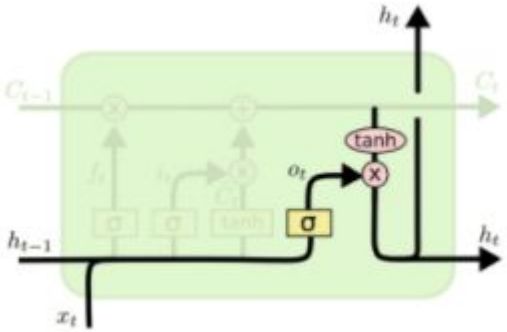
😊 Gated Recurrent Neural Networks

1. Long Short-Term Memory Unit

RNN과 달리 tanh함수와 sigmoid 함수를 통해

$$h_t^j = o_t^j \tanh\left(c_t^j\right)$$

값의 크기를 일정 수준으로 유지한다.

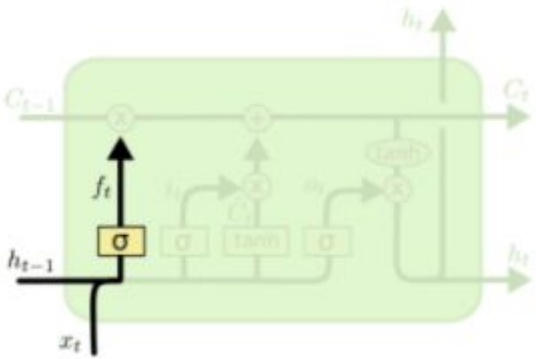


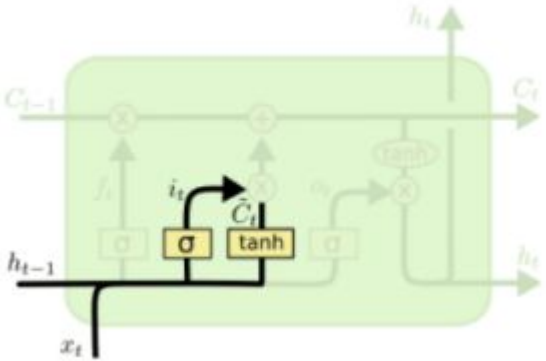
$$o_t^j = \sigma\left(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + V_o \mathbf{c}_t\right)^j$$

😊 Long Short Term Memory Unit (2)

description of creating new memory cell

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j \qquad \tilde{c}_t^j = \tanh\left(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1}\right)^j$$





😊 Long Short Term Memory Unit (3)

what we calculate in forget gate and input gate

$$
\begin{aligned}
f_t^j &= \sigma\left(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + V_f \mathbf{c}_{t-1}\right)^j \\
i_t^j &= \sigma\left(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + V_i \mathbf{c}_{t-1}\right)^j .
\end{aligned}
$$

😊 LSTM Summary

- overwriting 하는 방식의 RNN과 다르게, 기존 정보를 유지할지 안할지를 Gate를 통해서 결정한다.

- 중요한 feature가 발견되면, 더 길게 정보가 유지되고 long term dependency problem을 해결 할 수 있습니다.

😊 Gated Recurrent Neural Networks

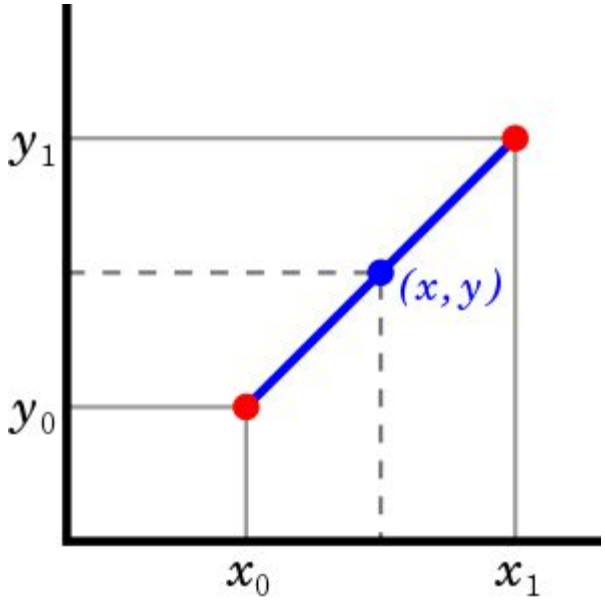
2. Gated Recurrent Unit

- memory cell 이 없다.
- ht를 구하는데, Linear interpolation (선형 보간법) 이 사용되었다.

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

😊 What is Linear interpolation?

- 양 끝점이 주어졌을 때,
- 그 사이에 위치한 점을 추정하는 방법




```
// p1,p2를 d1:d2로 분할하는 p를 리턴한다. (단, d1+d2=1)
float lerp(float p1, float p2, float d1) {
  return (1-d1)*p1 + d1*p2;
}
```

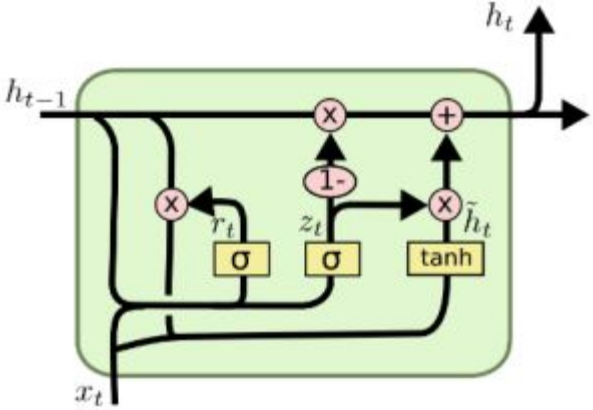
https://ko.wikipedia.org/wiki/%EC%84%A0%ED%98%95_%EB%B3%B4%EA%B0%84%EB%B2%95

😊 Gated Recurrent Unit (2)

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

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

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



😊 Gated Recurrent Unit (3) (with summary)

- Update gate : Content가 업데이트 된 정도

- reset gate : rt 값이 0에 가까워 질 때, 이전에 계산 되었던 모든 unit 값을 잊는다.

- candidate activation h는 RNN과 동일하게 동작한다.

😊 Discussion

새로운 기억 방식의 장점

1. 긴 series data 에서 각 unit 에서의 중요한 feature를 잘 기억한다.

2. forget gate, update gate에서 이전 정보 유지 정도를 결정하는 것이다.

3. 덧셈으로 인해 vanishing gradient problem을 해결해준다.

😊 Discussion (2)

두 unit의 차이점

1. LSTM에는 memory content 노출정도를 output gate에서 조절하지만, GRU에는 그런 부분이 존재 하지 않는다.

2. reset gate에 해당하는 input gate 의 위치가 다르다. 이에 따라 LSTM은 이전의 메모리의 총 양을 조절하는 것과 분리없이 새로운 memory content를 계산한다. 게다가, LSTM unit은 새로운 memory content의 총 양을 forget gate의 독립적인 cell과 더해져서 조절된다.

3. GRU는 이전의 연산에 의하여 정보의 흐름을 조절하지만, 독립적으로 정보의 총 양을 조절하지는 않는다.

😊 Models

- 세 모델을 근사하게 같은 param 수를 가지게 했다.

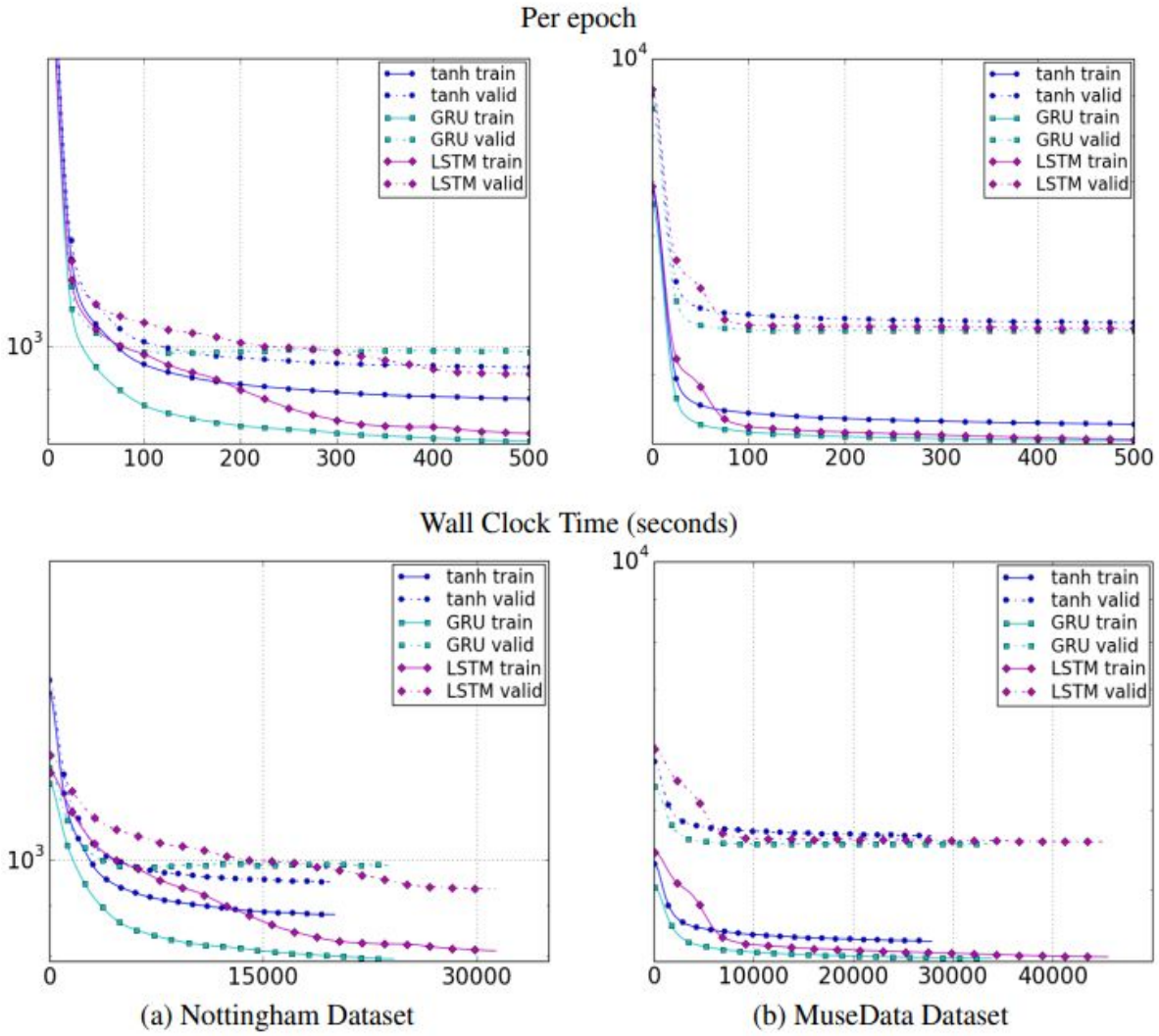
- overfitting을 피하기 위하여 적은 param 수를 사용했다.

Unit	# of Units	# of Parameters
Polyphonic music modeling		
LSTM	36	$\approx 19.8 \times 10^3$
GRU	46	$\approx 20.2 \times 10^3$
tanh	100	$\approx 20.1 \times 10^3$
Speech signal modeling		
LSTM	195	$\approx 169.1 \times 10^3$
GRU	227	$\approx 168.9 \times 10^3$
tanh	400	$\approx 168.4 \times 10^3$

Table 1: The sizes of the models tested in the experiments.

			tanh	GRU	LSTM
Music Datasets	Nottingham	train	3.22	2.79	3.08
		test	**3.13**	3.23	3.20
	JSB Chorales	train	8.82	6.94	8.15
		test	9.10	**8.54**	8.67
	MuseData	train	5.64	5.06	5.18
		test	6.23	**5.99**	6.23
	Piano-midi	train	5.64	4.93	6.49
		test	9.03	**8.82**	9.03
Ubisoft Datasets	Ubisoft dataset A	train	6.29	2.31	1.44
		test	6.44	3.59	**2.70**
	Ubisoft dataset B	train	7.61	0.38	0.80
		test	7.62	**0.88**	1.26

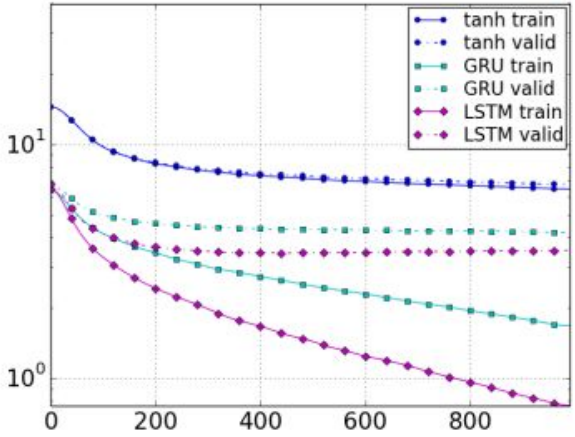
Table 2: The average negative log-probabilities of the training and test sets.

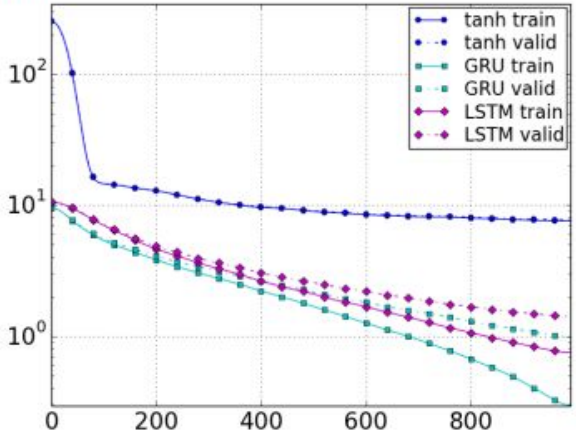


(a) Nottingham Dataset

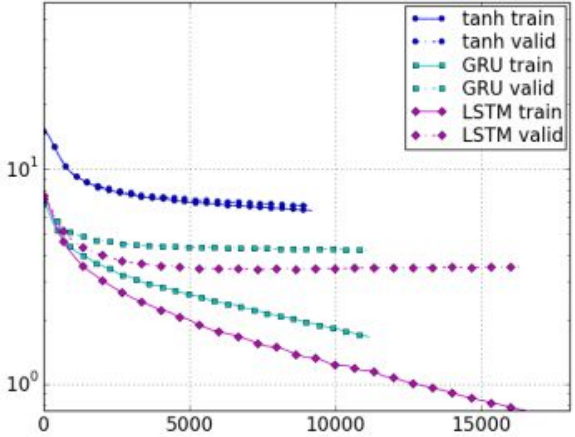
(b) MuseData Dataset

Per epoch

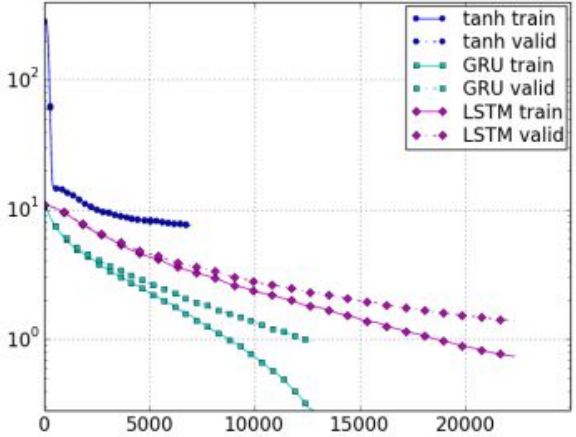




Wall Clock Time (seconds)



(a) Ubisoft Dataset A



(b) Ubisoft Dataset B

😊 Results and Analysis

- poly music dataset의 경우 GRU는 LSTM과 tanh-RNN보다 좋은 성능을 보여준다.

- music dataset에서는 세 모델의 성능 차이가 크지 않다.

- 음성인식 task의 경우 LSTM과 GRU의 성능이 좋았고, tanh-RNN은 좋지 못했다.

😊 Thanks for listening!