RNN + LSTM

Field	CV NLP Time Series
Journal	ICLR
# Published Year	1997
Speaker	나보영 조태완
Summary	RNN + LSTM
status	Finished!
Q&A	1. long term, short term 구분 2. 그레이디언트 소실문제 해결
link	https://www.bioinf.jku.at/publications/older/2604.pdf

A Learning Algorithm for Continually Running Fully Recurrent Neural Networks

Ronald J. Williams
College of Computer Science
Northeastern University
Boston, Massachusetts 02115

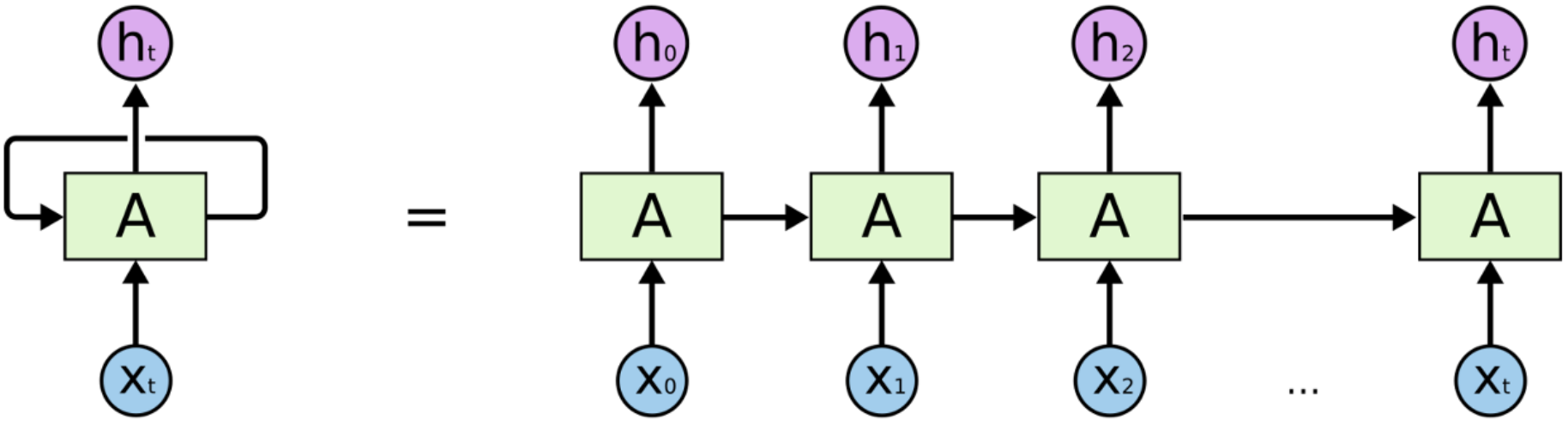
and
David Zipser
Institute for Cognitive Science
University of California, San Diego
La Jolla, California 92093

Appears in *Neural Computation, 1*, pp. 270-280, 1989.

Abstract

The exact form of a gradient-following learning algorithm for completely recurrent networks running in continually sampled time is derived and used as the basis for practical algorithms for temporal supervised learning tasks. These algorithms have: (1) the advantage that they do not require a precisely defined training interval, operating while the network runs; and (2) the disadvantage that they require nonlocal communication in the network being trained and are computationally expensive. These algorithms are shown to allow networks having recurrent connections to learn complex tasks requiring the retention of information over time periods having either fixed or indefinite length.

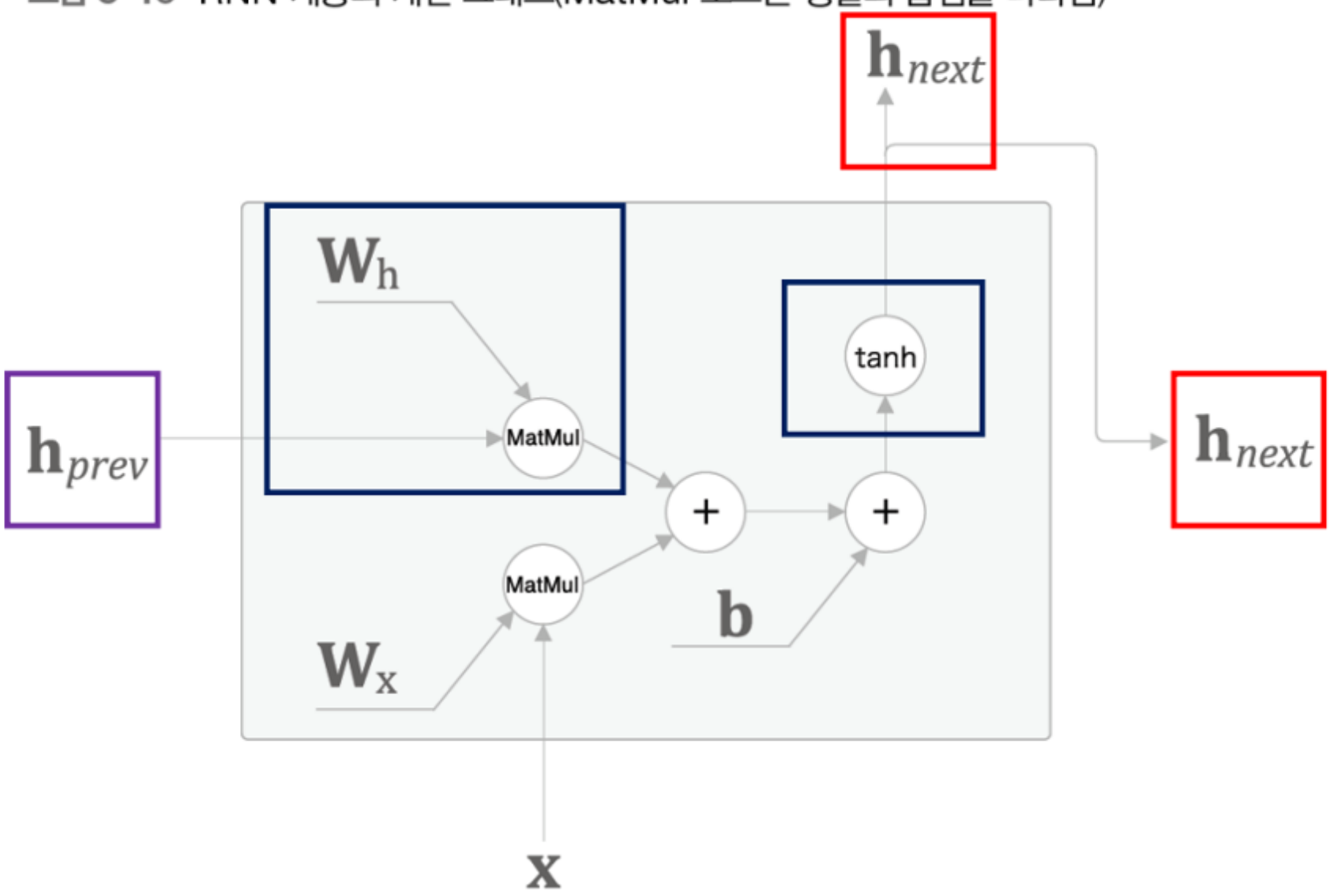
1. RNN 서론



- 시간에 따라 변하는 입력 또는 출력을 다루기 위해 만들어진 구조
- 재귀적 연결을 사용. 순차적인 정보를 처리

2. RNN 구조

그림 5-19 RNN 계층의 계산 그래프(MatMul 노드는 행렬의 곱셈을 나타냄)



- 이전 단계의 은닉 상태를 가져와 사용하는 방식으로 작동함
- 수도코드로 다음과 같이 나타낼 수 있습니다.

```
function RNN(inputs):
  hidden_state_t = INITIAL_STATE  // 보통 0으로 초기화
  total_hidden_states = EMPTY_LIST

  for input_t in inputs:
    output_t = tanh(Wx * input_t + Wh * hidden_state_t + b)
    append output_t to total_hidden_states
    hidden_state_t = output_t
```

```
  end for

  return total_hidden_states
end function
```

3. RNN의 한계

$$\left| f'_{l_m}(net_{l_m}(t-m)) w_{l_m l_{m-1}} \right| > 1.0$$

$$\left| f'_{l_m}(net_{l_m}(t-m)) w_{l_m l_{m-1}} \right| < 1.0$$

- 기울기 소실 문제, RNN의 Backpropagation 과정에서 1보다 작은 기울기가 계속 곱해지게 되면 0으로 수렴하게 되면서 발생합니다. 반대로 1보다 큰 기울기라면 무한대로 커져 기울기가 폭발하는 현상이 발생함.

4. RNN 결론

- 주된 목표는 “순환 알고리즘을 통해 시간에 따라 변화하는 임무를 지속적으로 업데이트 하는 것”
- 사전 정보 없이도 최적화되지 않은 훈련 데이터로부터 시간에 따른 작업을 처리할 수 있는 능력을 가짐
- 알고리즘이 병렬적이다(계산 과정에서 일부 병렬 사용 가능)
- 장기 메모리가 좋지 않다는 단점이 있다.

5. LSTM

LONG SHORT-TERM MEMORY

Neural Computation 9(8):1735–1780, 1997

Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

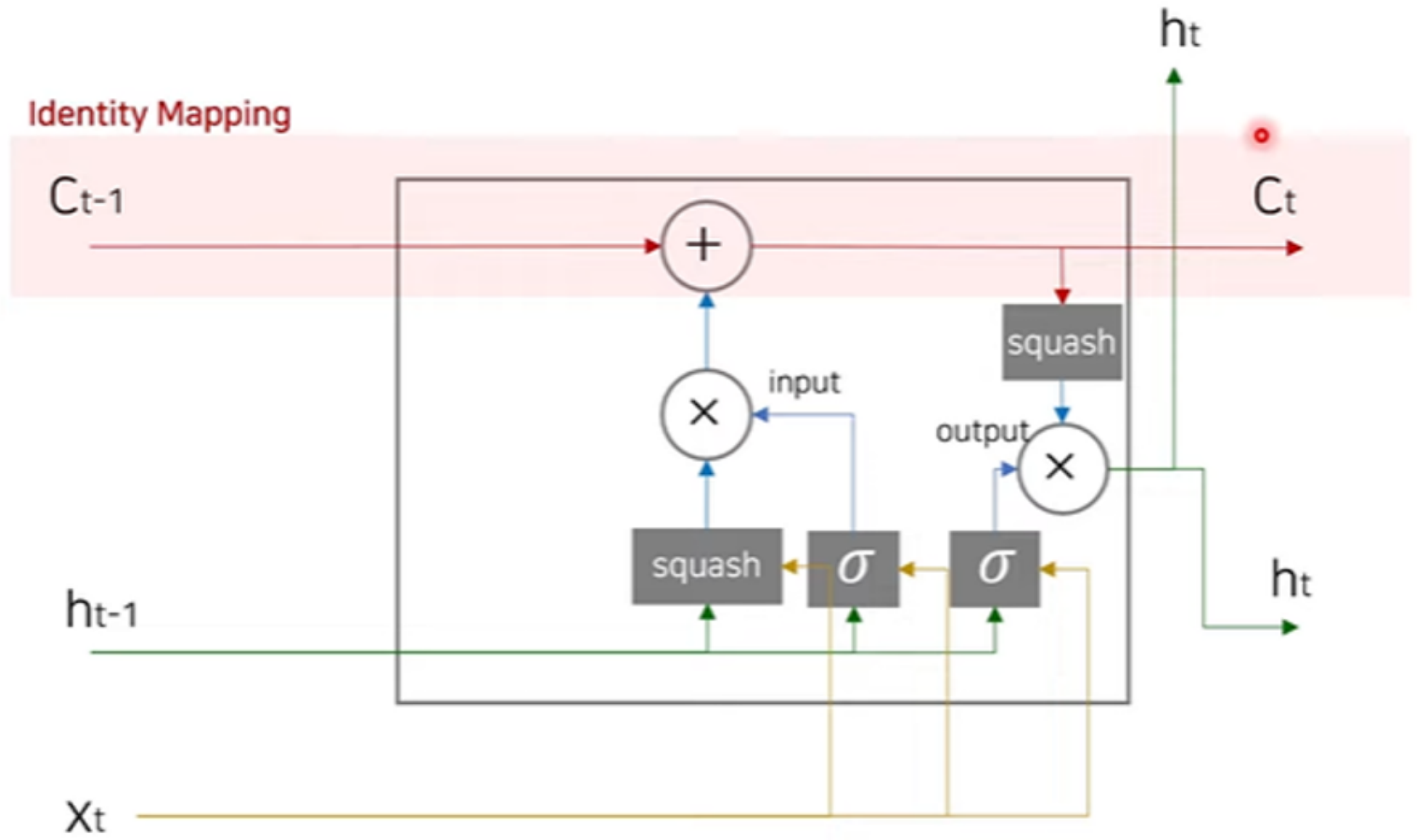
Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen

Abstract

Learning to store information over extended time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow. We briefly review Hochreiter's 1991 analysis of this problem, then address it by introducing a novel, efficient, gradient-based method called "Long Short-Term Memory" (LSTM). Truncating the gradient where this does not do harm, LSTM can learn to bridge minimal time lags in excess of 1000 discrete time steps by enforcing *constant* error flow through "constant error carrousels" within special units. Multiplicative gate units learn to open and close access to the constant error flow. LSTM is local in space and time; its computational complexity per time step and weight is $O(1)$. Our experiments with artificial data involve local, distributed, real-valued, and noisy pattern representations. In comparisons with RTRL, BPTT, Recurrent Cascade-Correlation, Elman nets, and Neural Sequence Chunking, LSTM leads to many more successful runs, and learns much faster. LSTM also solves complex, artificial long time lag tasks that have never been solved by previous recurrent network algorithms.

- 기존 RNN의 단점인 장기 기억력과 기울기 소실 또는 폭발 문제를 해결하기 위해 등장한 Long short-term memory

6. LSTM 구조



- 핵심은 장기 기억인 Cell State와, 단기 기억인 Hidden State가 있습니다.
- Squash함수는 출력 범위를 제한해 안정적으로 작동될 수 있도록 합니다.
- 수도 코드로 나타내면 다음과 같습니다.

```
function SimpleLSTM(inputs):
  state.h = INITIAL_HIDDEN_STATE  // 보통 0으로 초기화
  state.C = INITIAL_CELL_STATE  // 보통 0으로 초기화

  for input_t in inputs:
    // Concatenate previous hidden state with current input
    concat_input = concatenate(state.h, input_t)

    // Compute input gate
    i_t = sigmoid(W_i * concat_input + b_i)

    // Compute candidate cell state (same as in basic RNN)
    C_tilda = tanh(W_C * concat_input + b_C)

    // Compute new cell state (without forget gate)
    C_t = state.C + i_t * C_tilda

    // Compute output gate
    o_t = sigmoid(W_o * concat_input + b_o)

    // Compute new hidden state
    h_t = o_t * tanh(C_t)

    // Update the state
    state.h = h_t
    state.C = C_t
  end for

  return state
end function
```

7. LSTM 평가

Method	Delay p	Learning rate	# weights	% Successful trials	Success after
RTRL	4	1.0	36	78	1,043,000
RTRL	4	4.0	36	56	892,000
RTRL	4	10.0	36	22	254,000
RTRL	10	1.0-10.0	144	0	> 5,000,000
RTRL	100	1.0-10.0	10404	0	> 5,000,000
BPTT	100	1.0-10.0	10404	0	> 5,000,000
CH	100	1.0	10506	33	32,400
LSTM	100	1.0	10504	100	5,040

Table 2: *Task 2a: Percentage of successful trials and number of training sequences until success, for "Real-Time Recurrent Learning" (RTRL), "Back-Propagation Through Time" (BPTT), neural sequence chunking (CH), and the new method (LSTM). Table entries refer to means of 18 trials. With 100 time step delays, only CH and LSTM achieve successful trials. Even when we ignore the unsuccessful trials of the other approaches, LSTM learns much faster.*

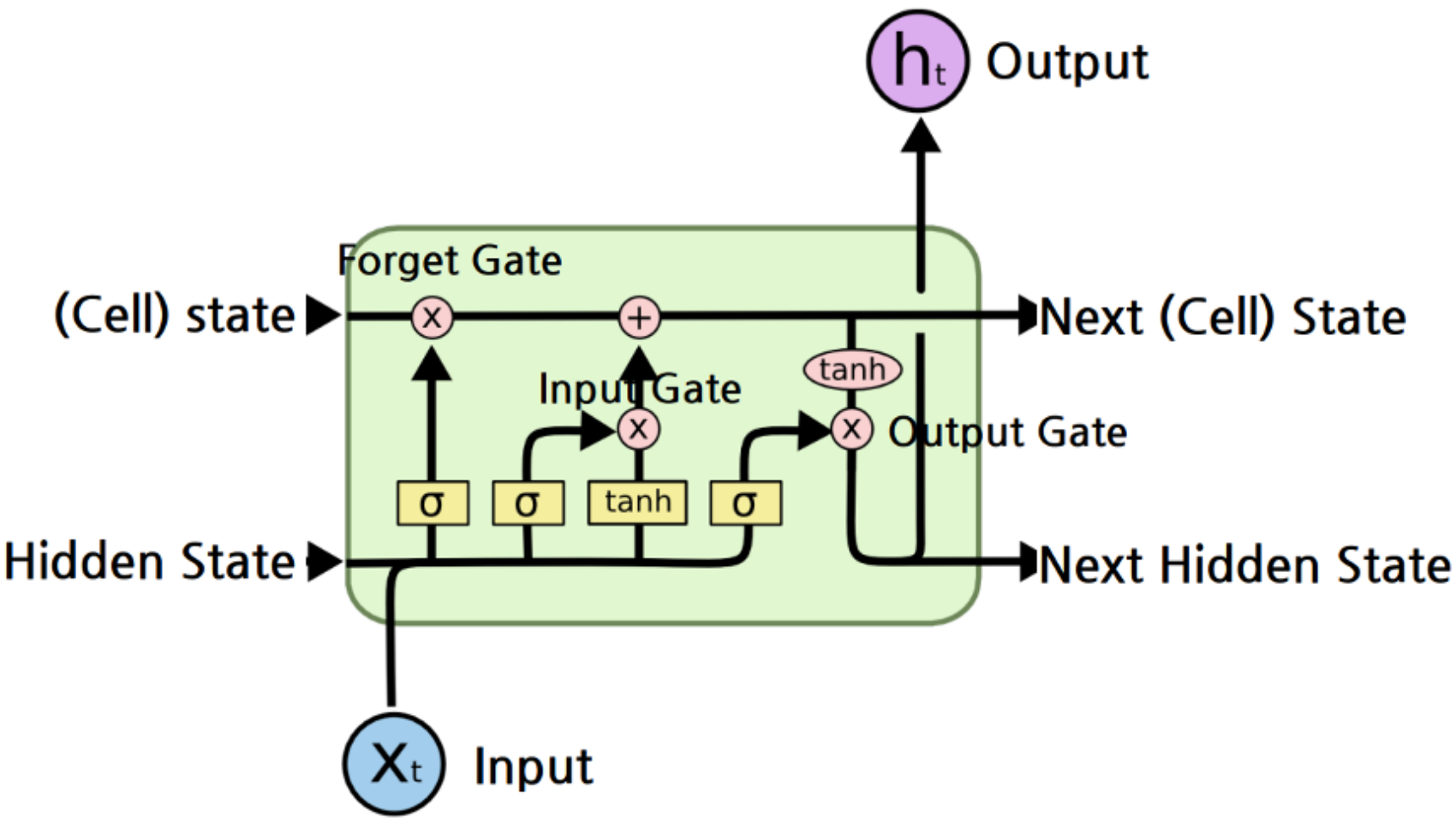
- Noise-free sequences with long time lags
- 이 실험은 LSTM이 긴 시간 지연을 가진 시퀀스 데이터에서 어떻게 성능을 발휘하는지를 보여주기 위한 것입니다.
- 따라서 평가 과정에서는 장기기억을 담당하는 Cell State와 입력 게이트만 사용했습니다.
- P가 100 이상일 때, LSTM 모델만이 100% 확률로 성공하는 것을 볼 수 있습니다. 이 결과는 LSTM이 기존의 CH 모델, RTRL, BPTT와 비교해 긴 시간 지연에서도 안정적이고 정확하게 작동함을 보여줍니다.
- CH, RTRL, BPTT는 모두 RNN 변형이나 학습 알고리즘이지만 Cell State라는 장기 기억력을 가지고 있지 않습니다.

T	minimal lag	# weights	# wrong predictions	Success after
100	50	93	1 out of 2560	74,000
500	250	93	0 out of 2560	209,000
1000	500	93	1 out of 2560	853,000

Table 7: *EXPERIMENT 4: Results for the Adding Problem. T is the minimal sequence length, $T/2$ the minimal time lag. "# wrong predictions" is the number of incorrectly processed sequences (error > 0.04) from a test set containing 2560 sequences. The rightmost column gives the number of training sequences required to achieve the stopping criterion. All values are means of 10 trials. For $T = 1000$ the number of required training examples varies between 370,000 and 2,020,000, exceeding 700,000 in only 3 cases.*

- Adding Problem: 시퀀스 데이터에서 특정 조건을 만족하는 요소들을 합산하는 문제
- 각 입력 시퀀스는 두 부분으로 구성된 Pair들로 이루어져 있으며, Pair의 첫 번째 요소는 [−1,1] 범위의 랜덤 실수, 두 번째 요소는 [1.0,0.0,−1.0] 중 하나입니다. 출력은 두 번째 요소가 1.0인 모든 Pair의 첫 번째 요소의 합이며, 성공 기준은 오류가 0.04 미만인 경우입니다.
- 이 실험 설계는 랜덤성과 복잡성을 고려하여 LSTM의 일반화 능력과 복잡한 패턴 학습 능력을 측정합니다. 특히, 두 번째 요소가 1.0인 Pair의 첫 번째 요소를 식별하고 합산하는 과정은 긴 시간 지연과 복잡한 상호작용을 필요로 하기 때문에, LSTM이 이를 얼마나 잘 처리하는지를 평가하는 데 아주 유용합니다.

8. LSTM의 발전



- 이후2년 뒤인 1999년 Learning to Forget: Continual Prediction with LSTM논문에서 LSTM알고리즘에 forget gate 를 추가시키면서, 장기 의존성과, 연산의 효율이 더 증가했고, 다양한 분야에 활용 할 수 있도록 진화하였습니다.
- Forget gate는 메모리 셀에서의 정보 유지와 삭제를 결정하는 역할을 하며, 이로써 LSTM이 과거 정보를 얼마나 잊을 것인지 또는 유지할 것인지를 세밀하게 제어할 수 있게 되었습니다.

9. LSTM 결론

- LSTM(Long Short-Term Memory)은 기존의 RNN(Recurrent Neural Network) 구조를 개선하여 long-term dependencies를 효과적으로 학습할 수 있게 만든 알고리즘입니다. LSTM은 메모리 셀과 여러 게이트 메커니즘을 도입하여 그레이디언트 소실 문제를 해결하고, 긴 시간 지연을 가진 데이터에서도 안정적으로 학습할 수 있게 하였습니다.
- 향후 연구에서는 이러한 LSTM의 능력을 더 다양한 실제 세계의 문제에 적용할 수 있을 것이라는 기대가 있습니다. 특히, "Sequence Chunkers"와 같은 다른 모델과의 결합을 통해, LSTM이 어떻게 다양한 시퀀스 처리 작업에서 장점을 발휘할 수 있는지에 대한 연구가 이루어질 것입니다. 이러한 연구는 LSTM의 다양한 응용 분야와 그 장점을 더 깊게 이해하는 데 도움을 줄 것으로 기대됩니다.

참고자료

rnn+ lstm 수정.pptx

https://colab.research.google.com/drive/1sARz2Uu4GY2baoMTIOcrMyJ-N862mZo0#scrollTo=F0vbGos7YaEd

https://wikidocs.net/22886

https://ratsgo.github.io/natural language processing/2017/03/09/rnnlstm/

https://dgkim5360.tistory.com/179#comment13282826

https://velog.io/@nayeon_p00/딥러닝-모델-RNN-과-LSTM

https://cryptosalamander.tistory.com/175

https://www.youtube.com/watch?v=IgIHjiCgECw&t=759s

논문 주소

https://gwern.net/doc/ai/nn/rnn/1989-williams-2.pdf

https://www.bioinf.jku.at/publications/older/2604.pdf