유한 상태 오토마타의 추론을 위한 이차 순환 신경망의 학습 시간 단축

○
류수길, 강효진, 정현기, 정순호
부경대학교 전자계산학과

Reducing learning time of Second-order Recurrent Neural Network inferring Finite State Automata

○
Su-Kil Ryou, Hyo-Jin Kang, Hyeon-Ki Jung, Soon-Ho Jung
Dept. of computer science, Pukyong University

요 약

이차 순환 신경망을 사용한 유한 상태 오토마타 추론에서 학습 시간을 단축시키는 방법을 소개한다. 기존의 학습과정에서는 Error가 발생해도 전체 가중치들의 변화가 없는 즉, 학습이 이루어지지 않는 시점이 발생하는데도 불구하고 학습은 최대 epoch를 만날 때까지 진행되는 학습 시간의 낭비를 가져온다. 이런 비효율적인 학습과정의 원인을 소개하고 학습이 불가능한 시점에서 가중치를 다시 부여하여 새롭게 학습을 시작하는 개선된 방법을 기존 학습에 추가한다. 실험을 통해 개선된 학습 방법과 기존 학습 방법을 비교 분석한다.

1. 서 론

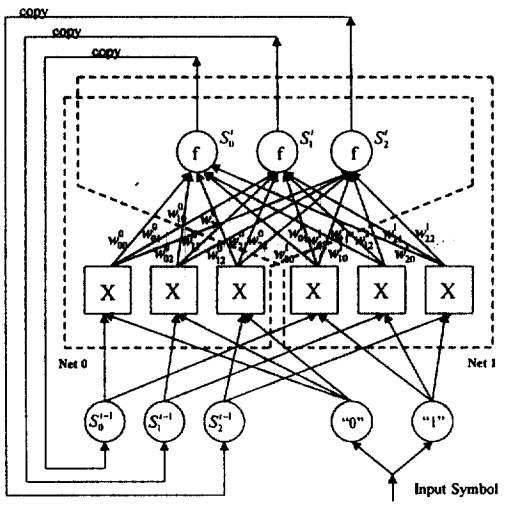
최근까지 신경망의 한 모델인 Recurrent Neural Network(RNN)을 사용하여 정규문법을 더욱 효율적으로 학습하기 위한 연구가 진행되어 왔다[1]-[4],[9]. 이들 연구에서의 RNN은 기존의 First-Order RNN(FRNN)에 비해 문법의 추론에서 더욱 성공적인 학습이 가능한 Second-Order RNN(SRNN)을 모델로 사용하게 된다[1]-[4],[6]. 여기서는 앞에서 소개한 SRNN 모델을 사용하고, 이 모델을 이용해 Finite State Automata(FSA)를 추론하게 된다[2],[5],[7]. 학습 알고리즘은 gradient learning 방법[10]이 쓰이며, 학습 패턴은 Tomita grammars로부터 생성된 예를 사용하고[1]-[5],[8], 입력 순서는 논문 [1]에서 제시한 바와 같이 positive 예와 negative 예를 혼합하여 길이에 의해 잘 정돈된 형태로 이 모델에 제공된다. 기존 학습 과정에서 학습 시간이 낭비되는 원인을 찾고, 분석 결과로 개선된 학습 방법을 소개한다.

2장에서는 기본 개념인 SRNN의 구조와 학습 방법을 살펴보고, 3장에서는 학습 종료시 가중치들의 분포와 기존의 학습 방법을 분석하여 학습 시간을 단축할 수 있는 방법을 소개하며, 4장은 실험으로써 기존의 학습 방법과 개선한 학습 방법으로 실험한 데이터를 비교하고 분석한다. 5장에서는 결론을 언급하고자 한다.

2. SRNN

앞서 설명한 내용과 같이 유한 상태 오토마타를 학습하기 위해 사용한 모델인 SRNN의 구조와 학습 방법인 gradient learning method을 살펴본다

2.1 Architecture



(그림 1) 상태수가 3인 SRNN

상태수가 3인 SRNN의 구조는 (그림 1)과 같이 보여지고, 입력 패턴은 "0"과 "1"로 이루어진 $x_0x_1x_2x_3\ldots x_L$ 구성되며, (그림 1)의 "X"는 두개의 입력인 "Input Symbol" unit과 상태 "S_i^{t-1}" unit의 곱(product)으로 출력된다. 따라서 현재 입력 x^t가 "1"이면 입력 unit "0"는 0값을 가지며, 입력 unit "1"은 1값을 가지므로 net1이 선택된다. 입력에 의해 선택된 net은 식(1)에 의해

$$S_i^t = h_i^t = f(\sum_j w_{ij}^{x^t} S_j^{t-1}) \qquad (1)$$

다음 상태 S_i^t가 계산되어 진다. 여기서 함수 f는 sigmoid function으로 아래와 같다.

$$f(x) = \frac{1}{1+e^{-x}}.$$

학습 과정은 논문 [2],[5]에서 소개하고 있다.

2.2 Gradient Learning Method

각 입력 pattern은 target 값과 함께 SRNN에 제공되며,. 각 입력 pattern의 끝(x^L)에서 error가 식(2)와 같이 계산된다.

$$E = \frac{1}{2}(h_0^L - T)^2 \qquad (2)$$

여기서, h_0^L는 식(1)에서 구할 수 있다.

만약 $|h_0^L - T| > \varepsilon$ (error tolerance)이면, 아래의 식(3)으로 가중치를 수정하게 된다.

$$w_{ij}^n = w_{ij}^n - \alpha \frac{\partial E}{\partial w_{ij}^n} \qquad (3)$$

이때 $\frac{\partial E}{\partial w_{ij}^n}$은 식 (1)을 미분한 식으로,

$$\frac{\partial E}{\partial w_{ij}^n} = (h_0^L - T)\frac{\partial h_0^L}{\partial w_{ij}^n} \qquad (4)$$

와 같고, 여기서 $\frac{\partial h_0^L}{\partial w_{ij}^n}$는

$$\frac{\partial h_k^t}{\partial w_{ij}^n} = f' \cdot (\sum_l w_{ij}^{x^t} \frac{\partial h_l^{t-1}}{\partial w_{ij}^n} + \delta_{ki}\delta_{nx^t} S_j^{t-1}) \qquad (5)$$

으로 계산된다.

성공적인 학습은 모든 패턴의 $|h_0^L - T|$가 ε보다 작은 값을 가질 때이고, 실패한 학습의 경우는 학습과정에서 최대 허용 Epoch에 도달할 때이며, 이것은 실험 값으로 설정된다.

3. SRNN 학습 과정과 개선 방안

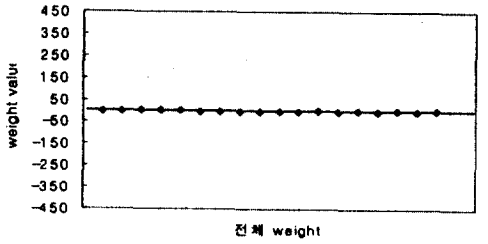
SRNN 학습과정을 실험을 통해 관찰된 최종 가중치들의 분포에 대한 분석과 알고리즘의 수리적인 분석을 통해 학습 과정에서의 문제점을 찾고 개선 방안을 제시한다.

3.1 SRNN 학습 과정의 분석

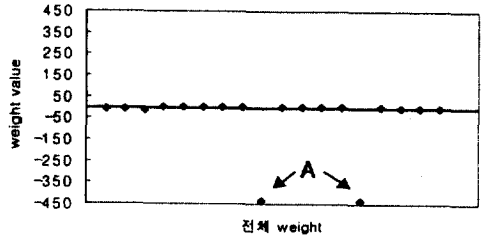
1) 기존 학습과정의 실험 분석

상태수가 3인 SRNN에서 gradient 학습 방법으로 실험을 했을 때 최종 가중치들의 분포 상태는 학습이 성공한 경우와 실패한 경우로 구분한다.

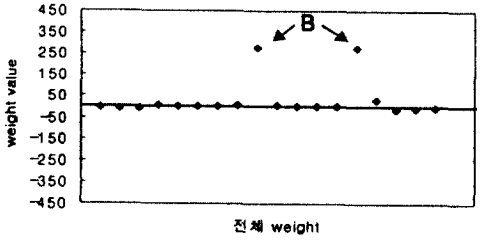
학습이 성공한 경우는 (그림 2)와 같이 나타나고, 가중치들은 0을 기준으로 균일하게 분포하는 특징을 발견할 수 있다. 그리고 학습이 실패한 경우는 크게 두 가지 분포상태를 가지는데, 첫 번째는 학습도중 최대 허용 epoch를 만나 학습이 실패하는 경우로 (그림 2)와 유사한 가중치 분포를 보인다. 두 번째는 학습과정에서 급격히 변화된 가중치들로 인해 이 후의 학습이 전혀 이루어지지 않는 경우로 (그림 3)과 같은 가중치 분포를 가진다. 위의 두 가지 분포에서 (그림 3)과 같은 형태가 실패한 경우에서 대략 80%정도 발생한다. 이런 경우 가중치의 값에 의해 두 가지 분포를 가진



(그림 2) 학습이 성공한 경우의 가중치들 분포



(a) 음의 방향으로 크게 이탈한 경우



(b) 양의 방향으로 크게 이탈한 경우

(그림 3) 학습이 실패한 경우의 가중치들 분포

다. (그림 3-a)는 몇몇 가중치들이 A처럼 음의 방향으로 크게 이탈하여 분포하고 있음을 보여주고, (그림 3-b)는 몇몇 가중치들이 B처럼 양의 방향으로 크게 이탈하여 분포하는 것을 보여준다. 따라서 몇몇 가중치들의 분포가 주요분포 지점으로부터 크게 이탈된 경우에는 학습이 실패한다. 이 원인은 학습과정에서 가중치 값의 변화과정에서 찾을 수 있었고, 그 결과 급격히 변한 가중치로 인해 Error가 발생하면서도 가중치 변화가 발생하지 않는 것을 알 수 있었다. 기존 학습 방법은 이런 상황에서도 최대 epoch까지 학습을 진행함으로써 학습 비용의 낭비가 발생하게 된다.

2) 기존 학습과정의 수리적 분석

학습알고리즘이 가지는 문제점을 수리적으로 표현하고자 한다. 먼저 주어진 패턴들을 학습할 경우 $|h_0^L - T|$가 Error 허용치(ϵ)보다 작은 경우와 큰 경우가 발생되는데, 이 둘 중에서 ϵ보다 큰 경우만을 고려하게 되며, 식(4)를 통해서 개선 방안을 찾을 수 있다. 식(4)는 $(h_0^L - T)$항과 $\frac{\partial h_0^L}{\partial w_{ij}^n}$항의 곱으로 계산된다.

위 식에서 Error가 발생한 경우는 아래와 같다.

$$| h_0^L - T | > \epsilon$$

이때 $\frac{\partial h_0^L}{\partial w_{ij}^n}$ 값에 의해 가중치의 변화량을 계산하게 된다.

가중치의 변화량 Δ는 식(4)로 부터 구할수 있다.

$$\Delta = \alpha (h_0^L - T) \frac{\partial h_0^L}{\partial w_{ij}^n}. \qquad (6)$$

① $\frac{\partial h_0^L}{\partial w_{ij}^n} = 0$ 일때.

식(6)에서 $\frac{\partial h_0^L}{\partial w_{ij}^n}$항이 0이면 $\Delta = 0$이 된다.

따라서 가중치들의 변화가 발생하지 않는다.

② $\frac{\partial h_0^L}{\partial w_{ij}^n} \neq 0$ 일때.

식(6)에서 α와 $(h_0^L - T)$는 0이 아닌 값이다.

따라서 $\Delta \neq 0$ 되고 정상적인 가중치의 변화를 수행하게 된다.

$\frac{\partial h_0^L}{\partial w_{ij}^n} = 0$이 되는 경우는 식(3)에서 모든 w_{ij}^n의 변화가 없게 된다. 이때 학습이 실패하는 원인을 좀 더 자세히 살펴보면 다음과 같다.

이 후의 식에서 $f'_{(k)} = f'(\sum_j w_{ij}^k S_j^{k-1})$ 고 한다.

$\frac{\partial h_0^L}{\partial w_{ij}^n} = 0$ 을 찾는 문제는 식(5)에서 $f'_{(L)} = 0$ 인 시점을 찾는 문제와 동일하고, 이것은 아래와 같이 풀어쓸 수 있다.

$$f'_{(L)} = f(\sum_j w_{ij}^L S_j^{L-1}) \cdot (1 - f(\sum_j w_{ij}^L S_j^{L-1}))$$

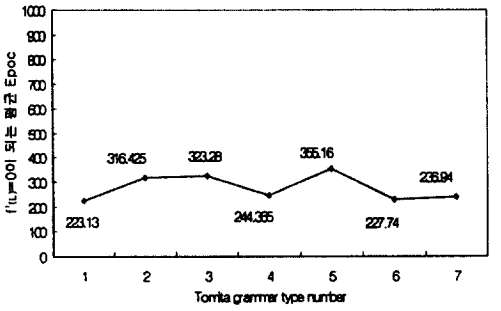
여기서 $f(\sum_j w_{ij}^L S_j^{L-1})$은 $|(\sum_j w_{ij}^L S_j^{L-1})| > \beta$ 일 때 0 또는 1의 값이 되며 이때 β는 대략 10정도의 값을 가진다. 이때 $f'_{(L)} = 0$이 되고, 이 시점이 발생할 때 학습이 이루어지지 않는다. 기존의 학습 과정은 이 시점을 고려치 않으므로 학습 시간이 낭비된다.

3.2 개선된 학습 과정

위에서 설명한 바와 같이 SRNN의 학습과정은 학습 패턴이 제공될 때 target과 output 사이의 차이(Error)를 가중치의 변화에 반영해야 하는데 $f'_{(L)} = 0$이 되는 상황이 발생하여 가중치들의 변화가 0이 된다. 이를 토대로 개선된 학습 방법을 소개한다. 이 방법은 Error가 발생하고 가중치의 변화가 0인 시점에서 가중치를 random한 값으로 새로 부여한 후 다시 학습을 시작하는 것이다. 새롭게 모든 가중치에 값을 부여하여 재학습하는 경우는 두 가지로 구분할 수 있다.

1) 학습과정에서 Error가 발생하여도 가중치의 변화가 0이 되는, 즉 $f'(\sum_j w_{ij}^{x^L} S_j^{L-1}) = 0$.

2) 학습이 최대 epoch에 도달



(그림 4) $f'(\sum_j w_{ij}^{x^L} S_j^{L-1}) = 0$인 평균 epoch수

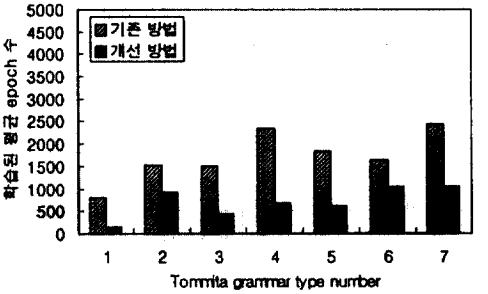
이 개선된 방법이 학습 속도를 줄일 수 있는 원인은 (그림 4)을 통해서 확인할 수 있다. 위의 (그림 4)에서 학습이 실패하는 시점은 학습 과정의 초기에서 발생되는 것을 알 수 있다. 따라서, 이 시점에서 가중치를 새롭게 부여하여 학습을 하게되면 학습시간이 훨씬 단축된다고 볼 수 있고 이를 실험을 통해서 확인하게 된다.

4. 실험

지금까지의 내용을 실험을 통해서 학습속도의 향상을 보이게 된다. 실험은 SRNN에서 기존의 학습 방법과 제안한 학습 방법을 Tomita grammars를 이용하여 실험하였고, 그 결과들로 비교 평가하고자 한다.

실험 과정을 설명하면 학습 패턴은 string길이가 5에서 15까지인 임의의 positive 예와 negative 예를 Tomita grammar 패턴 생성기를 통해 만들어 사용하고, 초기 가중치들은 -1과 1사이의 random한 값으로 초기화한다. 기존의 방법과 개선된 방법 모두 최대 epoch를 1000으로 하고 가중치의 재 할당은 5회로 제한한다. 기존 방법과 제시한 방법의 최대 epoch는 5,000으로 동일하므로, 학습에 걸리는 평균 epoch 수로써 학습 시간에 대한 평가를 할 수 있다.

(그림 5)는 Tomita grammar에서 기존 학습 방법과 개선된 학습 방법에서의 평균 epoch 수를 비교한 graph를 보여준다.



(그림 5) 학습된 평균 epoch 수

위의 실험 결과를 보면 개선된 학습 방법이 기존의 학습 방법보다 Tomita grammar 각각에 따라 최소 1.5배에서 최대 5배까지 학습 시간이 단축된다는 것을 알 수 있다.

5. 결 론

Positive 예와 Negative 예를 혼합하여 길이가 짧은 순에서 긴 순서로 정돈한 학습패턴을 SRNN의 입력으로 제공하여 Tomita grammar를 학습한다. 이 SRNN 학습과정은 Gradient learning algorithm에 의해 가중치를 변화시키게 되는데, 이때 Error가 발생해도 가중치의 변화가 0이 되어 학습이 이루어지지 않는 상황이 발생한다. 그럼에도 기존의 방법은 이런 상황을 고려치 않고 최대 허용 epoch까지 학습이 진행되는 시간적 낭비를 초래한다. 이런 비효율성을 줄이기 위하여 기존 학습방법을 최종 가중치의 분포에 대한 분석과 수리적인 분석을 통해 가중치의 변화가 0이 되는 시점을 감지하여 새로운 가중치를 다시 부여하는 개선된 학습방법을 설명한다. 실험을 통해서 개선된 방법이 기존 학습 방법보다 Tomita grammar에 따라 1.5배에서 5배까지 학습시간을 단축되는 것을 알 수 있었다.

따라서 유한 상태 오토마타의 추론을 위한 SRNN에서 학습속도를 줄이는 방법으로 정돈된 학습패턴의 입력순서[1]와 더불어 이 개선된 방법을 제안한다.

참고문헌

[1] 정현기, 정순호, "정규문법 추론을 위한 SRNN에서의 학습 집합의 입력 순서", 한국정보처리학회 '99 춘계 학술발표논문집, pp599-602, 1999.

[2] Richard J. Mammone, "Artificial Neural networks for Speech and Vision", Chapman & Hall, pp. 55-76, 1994.

[3] Z. Zeng, R. M. Goodman, P. Smyth, "Discrete Recurrent Neural Networks for Grammatical Inference", IEEE Trans. on Neural Networks, Vol. 5, No. 2, pp. 320-330, 1994.

[4] Z. Zeng, R. M. Goodman, P. Smyth., "Learning Finite State Machines With Self-Clustering Recurrent Networks", Neural Computation 5, pp. 976-990, 1993

[5] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, Y. C. Lee, "Learning and Extracting finite State Automata with Second-Order Recurrent Neural Networks", Neural Computation 4, pp.393-405, 1992.

[6] D. Angluin, "Inference of reversible languages", J.Assoc.Comput.Machin 29(3), pp.741-765, 1972.

[7] E. M. Gold, "System identification via state characterization", Automatics 8, pp.621-636, 1972.

[8] M. Tomita, "Dynamic construction of finite -state automata from examples using hill climbling", In Proceeding of the Fourth Annual Cognitive Science Conference, pp.105, 1982.

[9] T. Lin, B. G. Horne, "Learning Long-Term De pendencies in NARX Recurrent Neural Networks", IEEE Transaction on neural networks, VOL 7, NO 6, 1996.

[10] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult", IEEE Transaction on neural networks, VOL 5, NO 2, 1994.