인공 신경망이 학습한 지식을 이해하기 위한 규칙 시각화 도구

이은헌† · 김선빈† · 이헌주† · 김현철†
† 고려대학교 컴퓨터학과

Rule Visualization Tool for Understanding Knowledge of Trained Artificial Neural Network

Eun-Hun Lee† · Sun-bin Kim† · Hurn-joo Lee† · Hyeoncheol Kim†
† Dept. of Computer Science and Engineering, Korea University

요 약

오늘날 딥러닝은 교육을 포함한 다양한 분야에서 세상의 패러다임을 바꿀만큼 발전하고 있다. 그러나 딥러닝 모델이 어떤 지식을 습득하였는지 파악하기 어려워 딥러닝 시스템을 무조건적으로 신뢰할 수 없다는 것이 문제로 남아있다. 이 문제를 해결하기 위해 기존에 딥러닝이 학습한 결과를 If-then과 같은 형식의 규칙으로 추출하는 방법이 제안되었지만, 이러한 규칙은 사람이 이해하기에는 직관적이지 못하다는 단점을 가지고 있다. 본 논문에서는 이러한 문제를 해결하고자 딥러닝 모델이 습득한 지식을 규칙 형태로 추출하고 이를 시각화하여, 사람이 직관적으로 이해할 수 있는 형태로 표현하는 방법을 제시한다.

1. 서 론

최근 인공 신경망에 기반한 딥러닝은 그 우수한 성능으로 인해 다양한 분야에서 응용되고 있으며, 교육 분야 또한 이를 지능적 교육 시스템(ITS)에 응용하는 연구[1][2]가 진행 중이다. 그러나 딥러닝은 데이터로부터 어떠한 정보 혹은 지식을 학습하였는지 파악하기 어려운 블랙 박스(black box)의 문제를 가지고 있기 때문에 높은 정확도의 모델임에도 불구하고 이에 기반한 시스템을 무조건 신뢰하기 어렵다. 특히 교육 분야는 학생의 데이터를 통해 피드백을 제공해야할 필요성이 높은데, 딥러닝을 적용한 시스템은 데이터로부터 어떤 정보를 습득했는지 알 수 없기 때문에 정확한 피드백을 제공하기 어렵다.

기존에 딥러닝이 학습한 정보를 If-then 규칙이나 트리 형태 등 사람이 이해할 수 있는 형태로 바꾸는 방법들[3-7]이 연구되었다. 그러나 이러한 형태는 규칙이 복잡해질 경우 한 눈에 알아보기 어려운 형태이다. 본 논문은 규칙을 직관적으로 이해해야할 필요성에 착안하여, 딥러닝 모델이 학습한 지식을 시각적으로 표현하는 도구를 제안한다.

2. 이론적 배경

2.1 규칙 추출 알고리즘

신경망이 학습한 규칙을 추출하는 연구는 크게 3가지 접근법: decompositional 접근법, pedagogical 접근법, eclectic 접근법으로 분류할 수 있다.

Decompositional 접근법은 인공신경망의 내부 구조를 안다고 가정하고 규칙을 추출하는 접근 방법으로, 본 논문에서 사용한 KT 알고리즘[3][4]은 이에 기반한다. 반면에 pedagogical 접근법[5][6]은 인공신경망의 구조에 대한 이해 없이 오로지 입력 계층과 출력 계층과의 관계만을 통해서 학습된 규칙을 얻어내는 접근법이다. 마지막으로 eclectic 접근법[7]은 pedagogical 접근법과 decompositional 접근법의 절충적인 기법들을 포함하고 있다.

이 논문은 KT 알고리즘을 기반으로 하는데 이에는 두 가지 이유가 있다. 첫 번째로 pedagogical 접근법과 eclectic 접근법은 인공신경망의 내부 구조를 사용하지 않으므로 시각화에 부적합하기 때문이다. 신경망이 내부적으로 어떤 프로세스를 통해 규칙을 생성하였는지 알아야하기 때문에 이에 대한 정보 없이 규칙을 도출하는 접근법은 적절하지 않다. 두 번째로 decompositional 접근법을 사용한 알고리즘 중에서도 KT알고리즘은 퍼셉트론 구조를 변환 없이 직접적으로 사용하기 때문이다. 다른 decompositional 접근법은 가중치 값을 다른 구조 형태로 재구성하여 규칙을 추출하지만, KT 알고리즘은 퍼셉트론 구조를 온전히 보존하며 규칙을 추출하므로 규칙이 도출되는 과정을 시각화하기 알맞다.

2.1 신경망 시각화

신경망을 시각화를 하기 위한 연구로 Tzeng과 Ma [8]은 data-driven 시각화 모델을 제안하였다. [8]이 제안하는 신경망 시각화에서는 신경망의 입력과 출력 데이터 사이의 근본적인 의존성을 밝히기 위해서 블랙박스를 열기 위한 몇 가지 정보 시각화 디자인 실험을 진행하였다. 특히 이 실험들은 데이터의 시각화와 네트워크의 시각화를 결합하는 이점을 가지고 있었다.

2. 제안 모델

2.1 KT 알고리즘

KT 알고리즘[3][4]는 인공 신경망의 구조를 직관적으로 사용하는 규칙추출 알고리즘이다. 단일 퍼셉트론은 입력집합 I와 가중치 집합 W, 편향 b가 주어졌을 때, 이를 통해 활성화 함수 f가 임계값 세타를 넘는지 여부를 출력 O를 결정한다. 입력 집합의 부분집합 I'가 주어졌을 때 출력 O가 활성화 된다면, 부분집합 I'은 퍼셉트론을 활성화하는 규칙이라고 할 수 있다.

규칙에는 긍정 규칙(confirm rule)과 부정 규칙(disconfirm rule) 두 가지가 존재한다. 긍정 규칙은 출력 O를 활성화하는 규칙이다. 만약 0보다 큰 가중치를 가진 입력값이 출력을 활성시킨다면 해당 입력 값은 규칙을 긍정한다는 것을 의미한다. 반면 0보다 작은 가중치를 가진 입력값이 주어지지 않을 때 출력이 활성화 된다면 이는 해당 입력값이 규칙을 부정하는 것을 뜻한다. 이 규칙은 '$\mathrm{if}\ i_a \wedge \neg i_b \wedge \ldots \wedge i_z\ then\ O$'와 같은 'if-then'형식의 규칙으로 재작성될 수 있다. 부정 규칙은 긍정 규칙과 반대로 출력이 비활성화되는 규칙이다.

2.1.1 가지치기 방법

가능한 모든 조합으로부터 적합한 규칙을 찾는 것은 높은 정확도를 갖지만 복잡도가 높아지는 문제가 있다. 각 층에 n개의 입력이 존재하는 경우 총 2^n개의 규칙에 대한 조합이 가능성을 가지기 때문이다. m개의 은닉층 및 출력층이 있다고 가정한다면 전체 인공신경망의 생성할 수 있는 규칙 조합은 $m \bullet n \bullet 2^{n+1}$개이며, 즉 복잡도는 $O(n \bullet 2^N)$로 이는 NP-hard 문제에 속한다.

이 문제를 해결하기 위해 KT알고리즘은 다음과 같은 세가지 가지치기 방법을 제안하여 검색 공간을 줄였다. 첫 번째는 조합의 수를 k개로 제한하는 것이고, 두 번째는 KT알고리즘에서 입력집합의 속성을 pos-atts와 neg-atts집합으로 나눠 규칙을 추출하는 것이다. 마지막으로 3종류의 휴리스틱을 적용하여 검색공간을 줄이는 방법을 사용한다. 위에서 제시한 세 가지 방법은 모두 Big-O 관점에서 복잡도를 감소시키지 못하기 때문에 근본적인 해결책이 되지는 못한다.

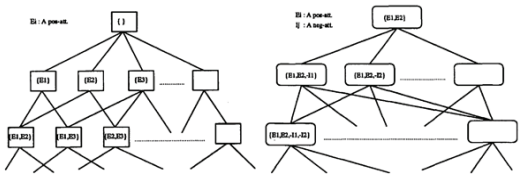

그림 1 explore-pos와 negate-neg의 규칙 탐색 공간

결론적으로 KT 알고리즘 접근법의 가장 큰 문제는 복잡도라 할 수 있으나, 검색공간을 줄이기 위한 세 가지 방법은 결과적으로 $O(n \bullet 2^n)$의 복잡도를 가지므로 근본적인 문제를 해결하지 못한다.

2.1.2 그래프를 그리기 위한 추가적인 함수

기존의 KT 알고리즘은 규칙이 추출되는 과정은 기록하지 않고, 현재 규칙의 요소를 하위규칙으로 교체할 뿐이다. 하지만 규칙의 추출 과정을 시각화하기 위해서는 규칙이 추출되는 과정을 기록해야한다.

규칙 추출 과정은 행렬을 이용한 방향성 그래프를 통해 기록하였다. 출력 노드에서 행렬의 각 행과 열은 순서대로 출력 노드, 은닉 노드, 그리고 입력 노드에 해당한다. 기록된 규칙 추출 과정에서 규칙이 교체 될 때, 그래프의 간선이 생성되었다는 의미를 기록하기 위해 행렬값을 0에서 1로 수정이 된다. 각각의 행렬은 한 가지 규칙에 매칭되며, 이 행렬을 기반으로 시각화 툴을 작성하였다.

2.2 시각화 모델

본 논문에서는 D3.js를 이용해 추출된 규칙과 신경망을 시각화 하는 툴을 개발하였다. 이 툴은 신경망과 추출된 규칙간의 관계를 시각화 한다. 이 시각화 툴의 구조는 그림 2를 참고하면 된다.

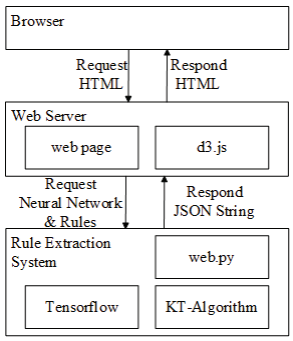

그림 2 시각화 툴의 구조

2.2.1 규칙 추출 시스템

TensorFlow는 구글이 개발한 머신러닝 시스템을 위한 오픈소스 라이브러리다. TensorFlow는 싱글 혹은

멀티 CPU와 GPU에서 동작을 한다. 본 논문에서는 TensorFlow를 신경망을 학습시키는데 사용하였다. 신경망을 학습시킨 뒤에는 신경망으로부터 KT 알고리즘을 사용하여 규칙을 추출하였다.

웹앱과 규칙추출 시스템 간에 통신을 위해 REST API을 사용하였는데, 이 REST API는 web.py를 이용해 구현을 하였다. Web.py는 파이썬에서 REST API를 구현할 수 있게 해주는 간단한 웹 프레임워크 이다.

규칙추출 시스템은 웹페이지로 부터 REST API를 통해 요청을 받으면 신경망의 학습을 수행하고 학습한 신경망으로부터 규칙을 추출한 다음 JSON형식으로 반환하도록 구현이 되었다.

2.2.2 시각화 툴

D3.js는 동적이고 상호작용하는 시각화를 웹브라우저에서 동작하도록 구현 할 수 있게 해주는 자바스크립트 라이브러리이다. 본 논문에서는 D3.js와 HTTP를 이용해 신경망과 추출된 규칙의 시각화 툴을 구현하였고, 'web.py'를 이용한 간단한 웹서버를 사용해 동작 테스트를 하였다.

3. 실험

본 논문에서는 모델이 제대로 작동하는 지 증명하기 위해서, 다음과 같은 실험을 준비하였다. 실험에는 간단한 XOR 데이터 집합이 사용되었다. KT 알고리즘의 복잡도가 높아 즉각적인 피드백을 구하기 어렵기 때문에 실질적인 데이터가 아닌 XOR 데이터 집합을 사용하여 실험을 진행하였다.

XOR를 훈련하기 위해 1개의 은닉층을 가진 인공신경망을 구성되었다. 모델은 2개의 입력노드와 4개의 은닉 노드, 그리고 2개의 출력노드를 갖고 있다. 이론적으로, 2개의 은닉 노드만으로도 XOR를 학습할 수 있있지만, 실제로는 2개의 은닉 노드만으로 XOR를 학습하는 것은 불가능에 가깝기 때문에, 4개의 은닉 노드를 사용하였다. 시그모이드 활성화 함수를 사용하고, 비용 함수로 cross-entropy를 적용하였으며, 신경망은 gradient descent 방식을 통해 학습되었다. Learning rate는 0.2로, 총 1000번의 epoch를 통해서 학습하였다.

KT 알고리즘의 변수 k 값이 작을 경우에는 조합이 너무 많이 제거되어 규칙이 제대로 생성되지 않고, k 값이 4인 경우에는 가지치기가 발생하지 않으므로, k 값을 3으로 설정하였다. 검색 공간이 너무 작아서, 임계값이 적용될 만한 규칙이 없기 때문에 각각의 휴리스틱에 대한 임계값은 모두 0으로 설정하였다.

4. 결과

본 논문의 모델은 웹에서 가장 많이 사용되는 시각화 도구인 D3.js를 사용하여, 훈련된 인공신경망을 시각화하였다. 신경망이 학습한 규칙은 노드와 간선으로 구성된 그래프의 형태로 나타내어진다.

시각화 모델은 노드와 간선들의 색상을 통해서, 추출된 규칙을 시각화한다. 몇 초마다, 바뀌는 색상들을 통해서, 추출된 긍정 규칙과 부정 규칙을 확인할 수 있다. 시각화 모델을 통해서 신경망이 학습한 규칙과, 신경망의 작동원리를 직관적으로 이해하고 파악할 수 있다. 즉각적인 인식을 위해서, 긍정 규칙을 표현하는 데는 빨간색을, 부정 규칙을 표현하는 데는 파란색을 사용하였다.

노드에 마우스를 올리면 그 노드가 활성화되기 위해서 활성화되어야 하는 노드들과 비활성화 되어야하는 노드들을 확인할 수 있다.

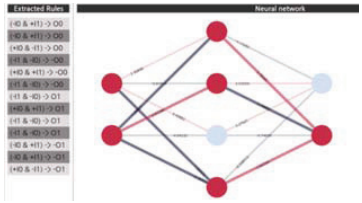

그림 3 전체 긍정 규칙 시각화

마우스로 노드를 가리키면, 노드들이 빨간색으로 변하는 것을 확인할 수 있다. 빨간색으로 칠해진 이 노드들은 마우스로 가리켜진 노드의 긍정 규칙을 의미한다. 빨간색 간선을 가진 빨간색 노드들은 마우스로 가리켜진 노드가 활성화되기 위해서 활성화되어야 한다. 반면에, 파란색 간선을 가진 빨간색 노드들은 마우스로 가리켜진 노드가 활성화되기 위해서 비활성화되어야 한다.

긍정 규칙을 나타내는 애니메이션이 끝난 뒤에는 부정 규칙을 확인할 수 있다. 부정 규칙은 파란색 노드로 시각화된다. 부정 규칙을 표현할 때는 간선의 색깔이 의미하는 바가 긍정 규칙을 표현할 때와 다르다. 긍정 규칙을 표현할 때, 마우스가 가리키는 노드가 활성화되기 위해서 활성화되어야 하는 노드는 빨간색 간선을 가진 빨간색 노드로 표현된다. 그러나 부정 규칙을 표현할 때는 같은 의미를 파란색 간선을 가진 파란색 노드들로 표현한다. 또한, 부정 규칙을 표현할 때는, 빨간색 간선을 가진 파란색 노드를 통해서 마우스로 가리켜진 노드가 활성화되기 위해서 비활성화 되어야 하는 노드들을 나타낸다.

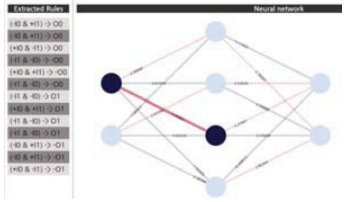

그림 4 단일 부정 규칙 시각화

규칙 시각화는 한 계층의 단계에 국한되는 것이 아니라 전 계층의 규칙에 대해서 시각화하였다. 즉, 출력계층에 대한 규칙뿐만 아니라, 은닉 노드에 대한 규칙들까지 모두 시각화하였다. 그림 7은 신경망의 긍정 규칙의 전체 단계를 시각화한 것을 나타낸 그림이다.

웹 사이트의 왼쪽에 위치한 추출된 규칙을 마우스로 가리키면, 출력노드를 활성화시키기 위한 전체 단계를 볼 수 있다.

5. 결론 및 향후 방향

이 논문은 2가지 방향에서 기존의 문제점을 개선하고 있다.

첫째로 기존의 규칙 집합을 규칙의 전체 과정을 나타낼 수 있는 그래프 형태로 변환하였다. 기존의 규칙 생성 프로세스를 통해서 나온 규칙 집합들은 중간 과정 즉, 은닉노드에 대한 규칙이 남아있지 않고 출력노드가 활성화되기 위해 필요한 입력노드들에 대한 규칙만이 남아있다. 그렇기 때문에, 사라지는 은닉 노드들의 규칙들을 시각화하기 위해서 그래프 구조를 설계하였다. 이를 통해서, 은닉계층의 노드들의 규칙들을 보존함으로써, 규칙의 전체 과정을 시각화할 수 있었다.

두 번째로 신경망이 학습한 규칙에 대한 직관적인 경험을 위해 규칙을 시각적으로 나타냈다. 인공신경망 전체의 구조를 시각화할 뿐만 아니라, 각 노드별로 학습한 규칙을 시각화해주기 때문에, 직관적으로 특정 뉴런을 활성화시키기 위해서 어떤 뉴런들이 활성화되어야 하는지 파악할 수 있다.

모델의 향후 발전 계획은 크게 2가지이다. 첫 번째로 사용자와 상호작용할 수 있는 모델을 만드는 것이다. 웹 사이트를 통해서 사용자가 직접 신경망의 하이퍼 파라미터를 조정할 수 있도록 모델을 발전시킬 것이다. 두 번째로 좀 더 복잡한 모델에도 적용할 수 있게 발전시키는 것이다. 보통의 경우에는 은닉계층이 1개인 신경망을 사용하지 않기 때문에, 실제 상황에도 적용시킬 수 있도록 발전시켜야 한다. 이를 위해서는 알고리즘의 복잡도를 줄여야 하는데, 퍼셉트론의 구조를 직접적으로 사용하는 KT 알고리즘 대신에 그렇지 않고 좀 더 복잡도가 낮은 여러 알고리즘이 있다. 이런 종류의 알고리즘을 사용하여 규칙을 추출하고 시각화 방법을 제안할 수 있다면, 즉각적인 피드백을 제공할 수 있는 시각화 도구가 가능할 것이다.

Acknowledgement

" 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1A2B4003558)."

참 고 문 헌

[1] Kardan, A. A., Sadeghi, H., Ghidary, S. S., & Sani, M. R. F. (2013). Prediction of student course selection in online higher education institutes using neural network. Computers & Education, 65, 1-11.

[2] Naser, S. A., Zaqout, I., Ghosh, M. A., Atallah, R., & Alajrami, E. (2015). Predicting student performance using artificial neural network: in the Faculty of Engineering and Information Technology. International Journal of Hybrid Information Technology, 8(2), 221-228.

[3] Fu, L. (1991, July). Rule Learning by Searching on Adapted Nets. In AAAI (Vol. 91, pp. 590-595).

[4] Fu, L. (1994). Rule generation from neural networks. IEEE Transactions on Systems, Man, and Cybernetics, 24(8), 1114-1124.

[5] Thrun, S. (1993). Extracting provably correct rules from artificial neural networks. Sekretariat für Forschungsberichte, Inst. für Informatik III.

[6] Pop, E., Hayward, R., & Diederich, J. (1994). Rule-Neg: Extracting Rules from Trained ANN by Step-wise Negation. Neurocomputing Research Centre, School of Computing Science, Queensland University of Technology.

[7] Towell, G. G., & Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks. Machine learning, 13 (1), 71-101.

[8] Tzeng, F. Y., & Ma, K. L. (2005, October). Opening the black box-data driven visualization of neural networks. In Visualization, 2005. VIS 05. IEEE (pp. 383-390). IEEE.