



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년03월05일
(11) 등록번호 10-2224066
(24) 등록일자 2021년03월02일

(51) 국제특허분류(Int. Cl.)
G05B 23/02 (2006.01) G06N 3/08 (2006.01)
(52) CPC특허분류
G05B 23/0254 (2013.01)
G05B 19/04 (2013.01)
(21) 출원번호 10-2019-0133430
(22) 출원일자 2019년10월25일
심사청구일자 2019년10월25일
(56) 선행기술조사문헌
KR101929955 B1
KR1020190098107 A

(73) 특허권자
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
(72) 발명자
황형주
경상북도 포항시 남구 지곡로 155, 8동 1501호(지곡동, 교수아파트)
이은샘
전라북도 전주시 완산구 새터로 95, 107동 203호(서신동, 동아한일아파트)
(74) 대리인
특허법인이름리온

전체 청구항 수 : 총 7 항

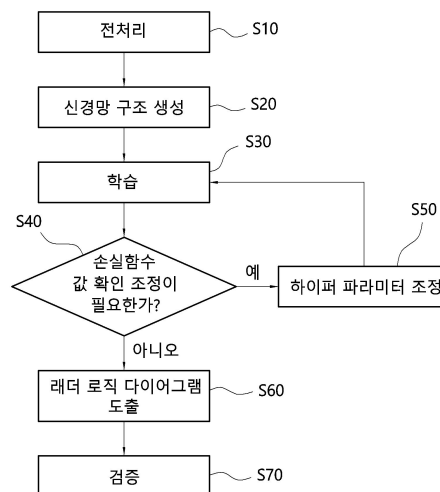
심사관 : 김윤한

(54) 발명의 명칭 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법

(57) 요약

본 발명은 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법에 관한 것으로, 컴퓨팅 장치를 사용하여 지능 제어기를 모델링하여 분석하는 방법으로서, a) 지능 제어기에서 사용되는 연산을 연속함수로 변환하여 입력하는 단계와, b) 은닉층, 선택층 및 함수층을 가지는 신경망 구조를 생성하는 단계와, c) 신경망 구조에 따라 학습을 수행하는 단계와, d) 학습의 손실함수값을 확인하여 조정이 필요한지 확인하는 단계와, e) 상기 d) 단계에서 조정이 필요한 것으로 판단되면, 상기 신경망 구조를 조정하고 다시 c) 단계를 수행하는 단계와, f) 상기 d) 단계의 판단결과 조정할 필요가 없으면, 래더 로직 다이어그램을 생성하는 단계를 포함할 수 있다.

대표도 - 도1



(52) CPC특허분류
G05B 23/0235 (2013.01)
G06N 3/08 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711089790
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구
연구과제명	딥러닝 알고리즘의 수학적 접근 및 AI 활용 금융기술 개발 연구
기 여 율	1/1
과제수행기관명	포항공과대학교
연구기간	2019.03.01 ~ 2020.02.29

명세서

청구범위

청구항 1

컴퓨팅 장치를 사용하여 지능 제어를 모델링하여 분석하는 방법으로서,

- a) 지능 제어기에서 사용되는 연산을 연속함수로 변환하여 입력하는 단계;
- b) 은닉층, 선택층 및 함수층을 가지는 신경망 구조를 생성하는 단계;
- c) 신경망 구조에 따라 학습을 수행하는 단계;
- d) 학습의 손실함수값을 확인하여 조정이 필요한지 확인하는 단계;
- e) 상기 d) 단계에서 조정이 필요한 것으로 판단되면, 상기 신경망 구조를 조정하고 다시 c) 단계를 수행하는 단계; 및
- f) 상기 d) 단계의 판단결과 조정할 필요가 없으면, 래더 로직 다이어그램을 생성하는 단계를 포함하는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

청구항 2

제1항에 있어서,

상기 손실함수는,

실제 출력값에 대한 평균제곱오차(Mean Squared Error, MSE)에 L^1 정규항(regularizing term)을 더한 값으로 설정되는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

청구항 3

제2항에 있어서,

상기 L^1 정규항의 파라미터는 0 이상의 실수인 것을 특징으로 하는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

청구항 4

제2항에 있어서,

상기 e) 단계는,

은닉층의 개수, 각 층 당 노드 개수 또는 상기 L^1 정규항의 파라미터를 조정하는 것을 특징으로 하는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

청구항 5

제1항에 있어서,

g) 상기 f) 단계에서 생성한 래더 로직 다이어그램의 입력값과 실제 지능 제어기의 입력값을 동일하게 입력하여, 양측의 출력값을 비교하여 검증하는 단계를 더 포함하는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

청구항 6

제1항에 있어서,

상기 b) 또는 c) 단계에서 가중치 행렬의 각 원소값을 0 또는 1의 값으로 충분히 가깝게 만들기 위해 가중치를

강화(Sharpening)기법을 이용하여 선택층을 구축하는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

청구항 7

제1항에 있어서,

상기 b) 단계의 함수층을 구성할 때 제어기의 연산에서 사용되는 연속함수를 입력하는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법.

발명의 설명

기술 분야

[0001] 본 발명은 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법에 관한 것으로, 더 상세하게는 기존의 지능 제어기를 컴퓨터 프로그램으로 모델링하고 숨겨진 작동 원리를 파악할 수 있는 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법에 관한 것이다.

배경 기술

[0002] 최근 스마트 공장 등 공장 자동화를 위하여 지능 제어기의 도입이 활발하게 이루어지고 있다. 지능 제어기는 프로그램 가능한 로직 컨트롤러(Programmable Logic Controller, PLC), 직접 계수 제어 장치(Direct Digital Controller, DDC), 분산 처리 시스템(Distributed Control System, DCS), 독립 실행형 컨트롤러(Standalone)를 포함하는 개념이다.

[0003] 공장 자동화를 위해서는 위의 제어기들의 유기적인 동작의 결합관계의 파악이 매우 중요하다. 하지만, 현재까지 국내에서는 각종 공정 제어 시스템을 구축할 때 해외 제조 업체로부터 장비를 수입하고, 수입된 설비들의 작동 원리나 유기적이 결합관계를 분석할 수 없기 때문에 설비의 유지 보수에 많은 비용을 지급하고 있다.

[0004] "Abdelhameed, Magdy M., Darabi, Houshang. Diagnosis and debugging of programmable logic controller control programs by neural networks. Proceedings of the 2005 IEEE International Conference on Automation Science and Engineering, Edmonton, Canada, August 1 & 2, 2005, ISBN: 0-7803-9426-7, 313-318 쪽"에는 순환 신경망을 이용하여 프로그램 가능한 로직 컨트롤러를 모델링하고 오류를 검출하거나 진단하는데 필요한 방안을 제시하였다.

[0005] 하지만 알고리즘 학습 과정의 추적이나 분석 과정이 결여되어, 컨트롤러의 내부 작동원리를 규명할 수 없으며, 다양한 알고리즘을 적용하지 않아 모델 정확도를 검증하기 어렵다는 한계가 존재한다.

[0006] "Prähofer, Herbert; Schatz, Roland; Wirth, Christian; Mossenbock, Hanspeter. A Comprehensive Solution for Deterministic Replay Debugging of SoftPLC Applications. IEEE Transactions on Industrial Informatics, Volume 7, Issue 4, November 2011, Article number 6009197, ISSN: 15513203, 641-651쪽", "Prähofer, Herbert; Wirth, Christian; Berger, Richard. Reverse engineering and visualization of the reactive behavior of PLC applications. IEEE International Conference on Industrial Informatics (INDIN), 2013, Article number 6622946, ISBN: 978-1-4799-0752-6, 564-571쪽", "Pirzadeh, Heidar; Agarwal, Akanksha; Hamou-Lhadj, Abdelwahab. An Approach for Detecting Execution Phases of a System for the Purpose of Program Comprehension. 2010 8th ACIS International Conference on Software Engineering Research, Management and Applications, Article number 5489832, ISBN: 978-076954075-7, 207-214쪽" 에서는 프로그램 가능한 로직 컨트롤 시스템의 추적 로그(trace log)를 이용하여 컨트롤러를 모델링하고 오류를 검출하는 방법을 제시하였다.

[0007] 하지만, 이와 같은 종래의 방법은 시스템 내부 작동 원리가 어느 정도 규명이 되어 있는 상태에서 모델링한다는 점에 있어서 큰 제약 조건을 가지고 있으며, 특히 추적 로그 정보를 얻을 수 없는 경우, 적용이 불가능하다는 단점이 있다.

[0008] 이와 같은 문제점을 고려하여 본 발명의 출원인의 등록특허 10-1929955호(2018년 12월 11일 등록, 인공지능 기반 역공학을 이용한 공장 자동화 시스템의 분석 방법)에서는 분석대상 지능 제어기의 입출력 데이터를 학습하여 논리모델을 구축하고, 내부 논리를 추출하는 방법에 대하여 설명하고 있다.

[0009] 그러나 위의 등록특허에 기재된 학습법은 트리 베이스(tree based) 모델을 이용한 것으로, 계층적 구조의 복잡성 때문에 목적에 따른 개량이 어려워, 유연성이 저하된다는 우려가 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명이 해결하고자 하는 기술적 과제는, 인공 신경망 알고리즘을 이용하여 지능 제어기를 분석할 수 있는 방법을 제공함에 있다.

[0011] 좀 더 구체적으로, 과도하게 중첩된 연산 때문에 내부 의사 결정 과정을 파악하기 어렵다고 알려진 인공 신경망 알고리즘으로 구축한 모델의 한계성을 해결하여 지능 제어기의 래더 로직 다이어그램을 복구 및 분석할 수 있는 방법을 제공함에 있다.

과제의 해결 수단

[0012] 상기와 같은 과제를 해결하기 위한 본 발명 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법은, 컴퓨팅 장치를 사용하여 지능 제어기를 모델링하여 분석하는 방법으로서, a) 지능 제어기에서 사용되는 연산을 연속함수로 변환하여 입력하는 단계와, b) 은닉층, 선택층 및 함수층을 가지는 신경망 구조를 생성하는 단계와, c) 신경망 구조에 따라 학습을 수행하는 단계와, d) 학습의 손실함수값을 확인하여 조정이 필요한지 확인하는 단계와, e) 상기 d) 단계에서 조정이 필요한 것으로 판단되면, 상기 신경망 구조를 조정하고 다시 c) 단계를 수행하는 단계와, f) 상기 d) 단계의 판단결과 조정할 필요가 없으면, 래더 로직 다이어그램을 생성하는 단계를 포함할 수 있다.

[0013] 본 발명의 실시예에서, 상기 손실함수는, 실제 출력값에 대한 평균제곱오차(Mean Squared Error, MSE)에 L^1 정규항(regularizing term)을 더한 값으로 설정될 수 있다.

[0014] 본 발명의 실시예에서, 상기 L^1 정규항의 파라미터는 0 이상의 실수일 수 있다.

[0015] 본 발명의 실시예에서, 상기 e) 단계는, 은닉층의 개수, 각 층 당 노드 개수 또는 상기 L^1 정규항의 파라미터를 조정할 수 있다.

[0016] 본 발명의 실시예에서, g) 상기 f) 단계에서 생성한 래더 로직 다이어그램의 입력값과 실제 지능 제어기의 입력값을 동일하게 입력하여, 양측의 출력값을 비교하여 검증하는 단계를 더 포함할 수 있다.

[0017] 본 발명의 실시예에서, 상기 b) 또는 c) 단계에서 가중치 행렬의 각 원소값을 0 또는 1의 값으로 충분히 가깝게 만들기 위해 가중치를 강화(Sharpening)기법을 이용하여 선택층을 구축할 수 있다.

[0018] 본 발명의 실시예에서, 상기 b) 단계의 함수층을 구성할 때 제어기의 연산에서 사용되는 연속함수를 입력할 수 있다.

발명의 효과

[0019] 본 발명 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법은, 인공 신경망 알고리즘을 이용하여 지능 제어기를 분석함으로써, 분석 목적에 따라 알고리즘의 변경이 용이하여, 유연성을 향상시킬 수 있는 효과가 있다.

[0020] 그리고 본 발명은 래더 로직 다이어그램으로 표현 가능한 제어기의 일부 논리 및 수식 연산에 대해서 가중치 강화법을 적용하여, 인공 신경망 알고리즘이 가지는 내부 의사 결정 과정을 용이하게 래더 로직 다이어그램으로 복구할 수 있는 효과가 있다.

도면의 간단한 설명

[0021] 도 1은 본 발명의 바람직한 실시예에 따른 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법의 순서도이다.

도 2는 본 발명에 적용되는 신경망의 일 실시 구조도이다.

도 3은 본 발명에 의해 생성된 래더 로직 다이어그램의 예시도이다.

도 4는 본 발명에서 사용되는 연속함수의 예시도이다.

도 5는 자동화 함수의 예시도이다.

도 6은 도 3의 래더 로직 다이어그램의 기능 포함하는 지능 제어기의 분석 결과 래더 로직 다이어그램이다.

발명을 실시하기 위한 구체적인 내용

- [0022] 이하, 본 발명 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법에 대하여 첨부한 도면을 참조하여 상세히 설명한다.
- [0023] 본 발명의 실시 예들은 당해 기술 분야에서 통상의 지식을 가진 자에게 본 발명을 더욱 완전하게 설명하기 위해 제공되는 것이며, 아래에 설명되는 실시 예들은 여러 가지 다른 형태로 변형될 수 있으며, 본 발명의 범위가 아래의 실시 예들로 한정되는 것은 아니다. 오히려, 이들 실시 예는 본 발명을 더욱 충실하고 완전하게 하며 당업자에게 본 발명의 사상을 완전하게 전달하기 위하여 제공되는 것이다.
- [0024] 본 명세서에서 사용된 용어는 특정 실시 예를 설명하기 위하여 사용되며, 본 발명을 제한하기 위한 것이 아니다. 본 명세서에서 사용된 바와 같이 단수 형태는 문맥상 다른 경우를 분명히 지적하는 것이 아니라면, 복수의 형태를 포함할 수 있다. 또한, 본 명세서에서 사용되는 경우 "포함한다(comprise)" 및/또는"포함하는(comprising)"은 언급한 형상들, 숫자, 단계, 동작, 부재, 요소 및/또는 이들 그룹의 존재를 특정하는 것이며, 하나 이상의 다른 형상, 숫자, 동작, 부재, 요소 및/또는 그룹들의 존재 또는 부가를 배제하는 것이 아니다. 본 명세서에서 사용된 바와 같이, 용어 "및/또는"은 해당 열거된 항목 중 어느 하나 및 하나 이상의 모든 조합을 포함한다.
- [0025] 본 명세서에서 제1, 제2 등의 용어가 다양한 부재, 영역 및/또는 부위들을 설명하기 위하여 사용되지만, 이들 부재, 부품, 영역, 층들 및/또는 부위들은 이들 용어에 의해 한정되지 않음은 자명하다. 이들 용어는 특정 순서나 상하, 또는 우열을 의미하지 않으며, 하나의 부재, 영역 또는 부위를 다른 부재, 영역 또는 부위와 구별하기 위하여만 사용된다. 따라서, 이하 상술할 제1 부재, 영역 또는 부위는 본 발명의 가르침으로부터 벗어나지 않고서도 제2 부재, 영역 또는 부위를 지칭할 수 있다.
- [0026] 이하, 본 발명의 실시 예들은 본 발명의 실시 예들을 개략적으로 도시하는 도면들을 참조하여 설명한다. 도면들에서, 예를 들면, 제조 기술 및/또는 공차에 따라, 도시된 형상의 변형들이 예상될 수 있다. 따라서, 본 발명의 실시 예는 본 명세서에 도시된 영역의 특정 형상에 제한된 것으로 해석되어서는 아니 되며, 예를 들면 제조상 초래되는 형상의 변화를 포함하여야 한다.
- [0027] 또한, 본 발명은 공장 자동화 등에 사용되는 지능 제어기의 입력과 출력 간의 관계를 이용하여 지능 제어기의 내부 로직을 파악하는 방법에 관한 것으로, 이러한 방법의 동작 주체는 통상의 연산이 가능한 제어기와 저장장치 및 데이터의 임시 저장 가능한 메모리를 포함하는 컴퓨팅 장치이며, 예를 들어 퍼스널 컴퓨터를 사용할 수 있다.
- [0028] 도 1은 본 발명의 바람직한 실시예에 따른 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법의 순서도이다.
- [0029] 도 1을 참조하면 본 발명은, 데이터를 전처리하는 전처리 단계(S10)와, 인공지능 학습을 위한 신경망 구조를 결정하는 신경망 구조 생성 단계(S20)와, 인공지능 학습을 통해 분석 모델을 생성하는 학습 단계(S30)와, 상기 학습 단계(S30)의 손실 함수값을 확인하여 조정이 필요한지 확인하는 확인 단계(S40)와, 상기 확인 단계(S40)에서 조정이 필요하다고 판단된 경우 하이퍼 파라미터를 조정하고, 상기 학습 단계(S40)를 재수행하는 조정 단계(S60)와, 상기 확인 단계(S40)에서 조정이 필요 없는 것으로 판단되면 래더 로직 다이어그램을 작성하는 래더 로직 작성단계(S60)와, 작성된 래더 로직을 검증하는 검증 단계(S70)를 포함한다.
- [0030] 이하, 상기와 같이 구성되는 본 발명의 바람직한 실시예에 따른 인공지능 기반 역공학을 이용한 지능 제어기의 분석 방법의 구성과 작용에 대하여 보다 상세히 설명한다.
- [0031] 먼저, S10단계인 전처리 단계에서는 분석 대상인 지능 제어기의 작동에 사용되는 연산을 확인하고, 각각을 수학적 연속 함수로 표현한다. 이는 사용자가 개입하여 컴퓨터를 이용하여 연속 함수를 입력하거나, 컴퓨터에 기저장된 연속 함수들을 선택하는 방식으로 수행될 수 있다.
- [0032] 전처리 단계(S10)는 경사 하강 기법을 사용하기 위해 필요한 과정이다.

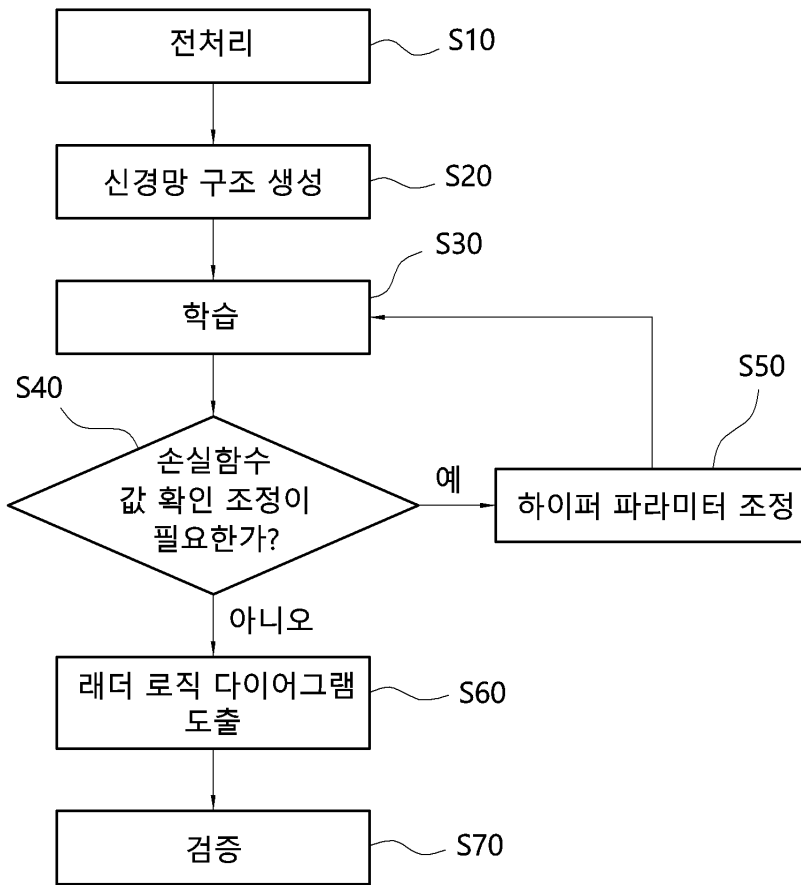
- [0033] 상기 연속 함수로 변환 가능한 기본 연산으로는 이산 값의 +, *, NOT, IDENTITY 연산 등을 예로 들 수 있다. 특히, 이산 값의 NOT 연산은 $1-x$ 로 표현 가능하며 두 개의 이산 값 x_1, x_2 의 + 연산은 $(x_1+x_2)-(x_1*x_2)$ 로 표현할 수 있다.
- [0034] 일반적으로 유한 개의 이산 값(discrete value)의 연산은 연속 함수로 근사 가능하다.
- [0035] 반면, 연속값(continuous value)을 인자로 갖는 계단 함수(step function)와 같이 불연속점이 존재하는 함수는 다른 연속 함수로 근사할 수 없다.
- [0036] 그 다음, 분석 대상인 지능 제어기의 입·출력 데이터를 신경망 알고리즘의 학습에 사용 가능하게 전처리하고, 입출력 데이터의 각 차원 수를 파악한다.
- [0037] 또한, 시간 지연(time delay)과 같은 연산자가 존재할 경우, 전처리 과정에서 이전 시점 데이터를 현재 시점의 입력값으로 취급하여 모델링할 수 있다.
- [0038] 이때, 과거 어느 시점의 데이터까지 입력값으로 취급할지 결정하는 할 수 있다.
- [0039] 그 다음, S20단계인 신경망 구조 생성 단계에서는 학습에 사용할 신경망 구조를 결정한다. 신경망 구조의 결정에 있어서 고려해야 하는 사항 중 하나는 구조 내에 은닉층(hidden layer)의 개수이며 다른 하나는 각 층에 필요한 노드(node)의 개수이다.
- [0040] 먼저, 은닉층을 결정하는데 필요한 요소는 문제의 복잡도이다. 문제가 복잡하면 할수록 이를 풀기 위해 많은 수의 은닉층이 필요하다. 이론적으로 문제 해결에 필요한 함수의 최소 합성 개수 이상의 은닉층이 있어야 알고리즘이 문제를 해결할 수 있다.
- [0041] 은닉층의 개수를 많게 설정하면 학습 가능성은 높아지지만 학습 이후 도출된 래더 로직 다이어그램이 불필요하게 복잡해질 가능성이 높다. 반대로 은닉층의 개수가 너무 적으면 학습이 안 될 가능성이 있다.
- [0042] 각 층에 필요한 노드(node)의 개수는 각 은닉층에 할당할 연산의 종류와 개수에 따라 결정된다. 예를 들어, 특정 은닉층에 단변수 연산을 m 개, 2변수 연산을 n 개 할당한다면 노드의 개수는 적어도 $m+2n$ 개여야 한다.
- [0043] 각 층별 노드의 개수를 결정하는데 중요한 요소는 연산의 중복 사용 빈도이다. 연산을 많이 중복하여 사용할수록 노드의 개수는 늘어나고 이에 따른 학습 가능성은 높아지게 된다. 하지만, 너무 많은 중복을 허용하면 학습 속도가 느려지는 문제가 발생한다.
- [0044] 실제로는 주어진 데이터에 대해 문제의 복잡도를 사전에 알 수 없으므로 은닉층과 노드의 개수를 구하는 방안은 정해진 것이 없으며, 이를 고려하여 선택한 후 시행착오를 거쳐 여러 가지 방안 중 최적을 선택해야 한다.
- [0045] 신경망의 각 은닉층은 선택층(selection layer)과 함수층으로 구성되어 있다.
- [0046] 도 2는 본 발명에 적용되는 신경망의 일 실시 구조도이다.
- [0047] 도 2를 참조하면, 은닉층(12, 22)이 두 개이고 각 층별 노드가 6개(단변수 연산 2개, 2변수 연산 2개 사용)인 신경망을 도시하였다.
- [0048] 선택층(11, 21, 31)에서는 가중치 행렬곱을 사용하여 연산을 적용할 변수를 할당하며, 함수층(13, 23)에서는 선택층에서 할당된 변수들이 주어진 연산들의 입력값으로 들어가게 된다.
- [0049] 최종 출력값의 차원을 맞추기 위해 신경망의 마지막 층은 선택층(31)으로 구성한다.
- [0050] 학습을 위한 손실함수(loss function)값을 실제 출력값에 대한 평균제곱오차(Mean Squared Error, MSE)에 L^1 정규항(regularizing term)을 더한 형태로 설정한다.
- [0051] L^1 정규항은 손실함수가 극소값(local minimum)에 빠지는 것을 방지해 준다.
- [0052] 모델을 학습하기에 앞서 L^1 정규항의 파라미터를 설정해야 하는데 보통의 경우 이를 0과 5 사이의 값으로 설정할 수 있으나, 0 이상의 실수이면 적용이 가능하다.
- [0053] 이와 같이 신경망 구조를 생성한 후, S30단계인 학습 단계를 수행한다.
- [0054] 학습이 진행되면서 가중치가 업데이트되고, 손실함수값이 충분히 줄어드는지 여부를 확인한다. 예를 들어, 손실

함수값이 10^{-3} 이하로 떨어지면 학습이 잘 된 것으로 판단한다.

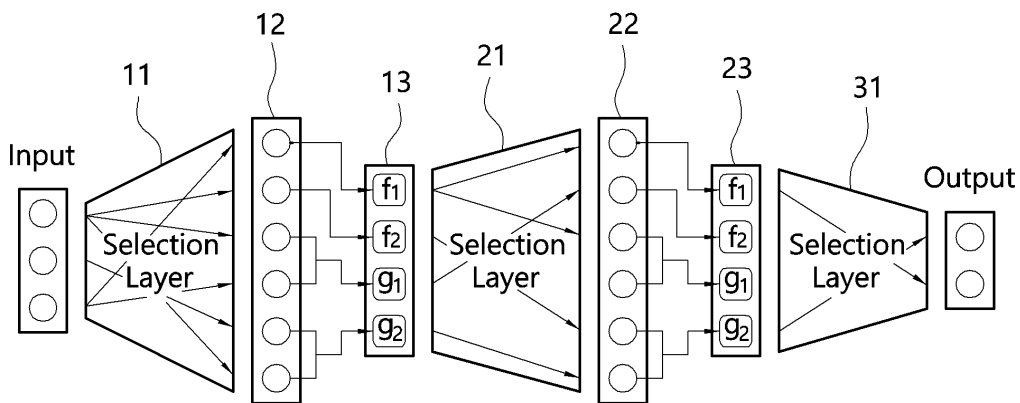
- [0055] 즉, S40단계에서는 손실함수값이 10^{-3} 이하인지 확인하고, 10^{-3} 를 초과하는 경우 S50단계의 하이퍼 파라미터 조정을 수행한 후 학습 단계를 재수행하며, 10^{-3} 이하인 경우 S60단계인 래더 로직 작성 단계를 수행한다.
- [0056] 래더 로직 다이어그램의 도출을 용이하게 하기 위해 학습 과정에서 중요한 부분은 가중치 행렬의 각 원소 값을 0 또는 1의 값으로 충분히 가깝게 만드는 것이다.
- [0057] 본 발명의 핵심 중 하나인 가중치를 강화(sharpening)하는 방안은 아래의 예시 중 하나를 택할 수 있다.
- [0058] 대표적인 예로 softmax를 취하는 것을 들 수 있다.
- [0059] 가중치를 ω 라고 했을 때, $\omega^y (y > 1)$ 에 softmax를 취하면 0 근처의 값을 더욱 0에 가깝게 만들 수 있다.
- [0060] 다른 방법으로, Gumbel softmax를 취하여 확률적으로 강화하는 것이 가능하다.
- [0061] 그 다음, S40단계에서는 앞서 설명한 바와 같이 손실함수값을 확인하여 학습이 잘되었는지 확인한다.
- [0062] 학습이 잘되지 않은 경우에는 S50단계인 조정 단계를 수행한다. 조정 단계는 하이퍼 파라미터를 조정하는 것으로, 기존의 학습을 중단하고 은닉층의 개수, 각 층 당 노드 개수, 정규항의 파라미터를 조정하여 다시 학습 단계(S30)를 수행한다.
- [0063] 상기 S40단계의 확인 결과 학습이 충분히 된 경우, 가중치 행렬의 각 원소 값은 0 또는 1에 가까운 값을 가지며 행렬에서 1에 가까운 값의 위치에 따라 적용된 연산과 변수의 종류가 무엇인지 결정된다.
- [0064] 따라서, S60단계인 래더 로직 작성 단계에서는 이 정보들을 역추적하는 방식으로 래더 로직 다이어그램을 도출한다.
- [0065] 이는 재귀 호출 함수를 이용한 프로그램으로 자동화할 수 있다.
- [0066] 도출된 래더 로직 다이어그램의 일례를 도 3에 도시하였다.
- [0067] 그 다음, S70단계인 검증 단계를 통해 작성된 래더 로직 다이어그램을 검증한다. 이는 앞서 학습에 사용된 모델의 정확성을 확인하는 것과 같다.
- [0068] 도출된 래더 로직 다이어그램을 이용해서 입력값과 출력값을 만들고, 실제 지능 제어기를 구동했을 때의 입력값 및 출력값과 비교하여 완전히 일치하는지 확인한다.
- [0069] 이하에서는 좀 더 구체적인 본 발명의 실시예에 대해 설명한다.
- [0070] 본 실시예는 본 발명을 이해하는데 초점을 맞춰 간단한 논리 연산을 수행한 입력 및 출력 모델을 선정하였으며, 이보다 복잡하거나, 매개 변수가 포함된 경우에도 동일한 방법으로 구현이 가능하다.
- [0071] 신경망 네트워크 모델 구축을 위한 공개 소프트웨어인 TensorFlow를 적용하였다. PyTorch로 대체하는 것도 가능하다.
- [0072] 간단한 로직에 대한 입·출력 데이터가 1000개가 주어져 있는 상황에서 입력변수의 개수는 5개, 출력변수의 개수는 1개가 있음을 가정한다.
- [0073] 이때 데이터로 사용된 정확한 래더 로직 다이어그램은 도 3의 래더 로직 다이어그램을 사용한다. 프로그램 상에서 입력값은 8차원으로, 출력값은 1차원으로 설정하였다.
- [0074] 입력값을 더 높은 차원으로 설정한 이유는 학습에 불필요한 입력값이 포함 된 경우에도 신경망이 결과를 잘 도출하는지 확인하기 위함이다.
- [0075] 연산은 +, *, NOT, IDENTITY가 주어져 있으며 이를 연속함수로 변환하여, 전처리 단계의 수행을 완료한다.
- [0076] 이때 연속함수의 예를 도 4에 도시하였다.
- [0077] 그 다음, 신경망 구조 생성 단계의 수행을 위해 은닉층의 개수는 3개로 설정하고, 각 은닉층마다의 연산은 단변수 연산인 IDENTITY를 2개, 2변수 연산인 +, *을 각각 2개씩 할당하였다. 따라서 노드 개수는 10개가 된다.
- [0078] 손실함수는 실제 출력값에 대한 평균제곱오차(MSE)에 L^1 정규항을 더한 형태로 설정했다.

도면

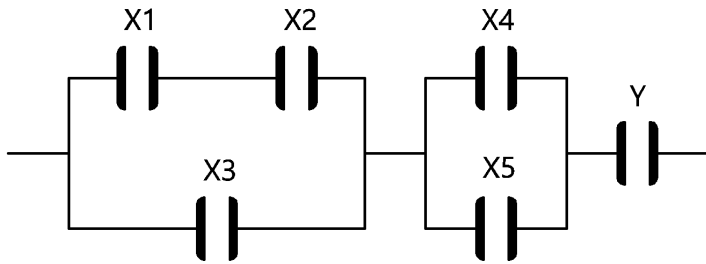
도면1



도면2



도면3



도면4

```

def Identity(x) :
    return x

def Not(x) :
    return 1-x

def Addition_logit(x1, x2) :
    return x1+x2-x1*x2

def Multiplication_logit(x1, x2) :
    return x1*x2
    
```

도면5

Expected Equation by WBN is
 $y_1 = ((X_3 \vee X_1) \wedge ((X_3 \vee X_2) \wedge (X_6 \vee X_4)))$

도면6

