



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

플럭스 코어드 와이어 설계 지원을 위한
탄력적 오류역전파 인공신경망의
적용성 검토

Applicability of Resilient Back-Propagation neural network
for support for Design of Flux-Cored Wire



지도교수 박 주 용

2018년 2월

한국해양대학교 대학원

조선해양시스템공학과

서 범 수

본 논문을 서범수의 공학석사 학위논문으로 인준함.

위원장 공학박사 남 중 호 (인)

위원 공학박사 조 효 제 (인)

위원 공학박사 박 주 용 (인)

2018년 2월

한국해양대학교 대학원

목 차

List of Tables	iii
List of Figures	iv
Abstract	vi

1. 서 론

1.1 연구배경 및 필요성	1
1.2 기존의 연구	2
1.3 연구의 목적	2

2. 플럭스 코어드 아크 용접(FCAW: Flux Cored Arc Welding)

2.1 FCAW 원리	3
2.2 용접 재료	4
2.3 용접 결과	7

3. 인공신경망

3.1 인공신경망의 분류 및 종류	9
3.2 인공신경망의 구조	10
3.3 퍼셉트론	12

4. 학습 방법

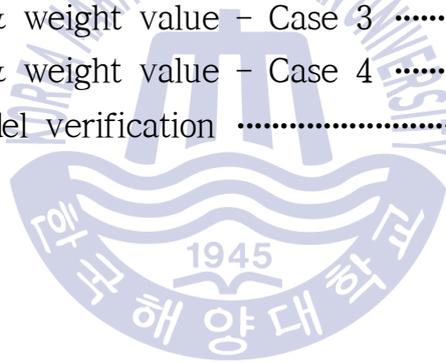
4.1 기존의 일반화된 델타 규칙	15
4.2 탄력적 오류역전파 알고리즘	22

5. 인공신경망을 이용한 용접재료설계 지원	
5.1 용접재료설계	27
5.2 용접재료설계지원 인공신경망시스템(WMANN)	28
5.3 인공신경망을 적용한 용접재료설계	37
6. 결론	
Reference	48



List of Tables

Table 1 XOR problem	13
Table 2 Characteristics of learning rule	32
Table 3 WMANN database table	36
Table 4 Learning rate & momentum used by algorithm	39
Table 5 Classify learning data	39
Table 6 Case of prediction model	40
Table 7 Detail factor of prediction model	40
Table 8 Connect No. & weight value - Case 1	43
Table 9 Connect No. & weight value - Case 2	44
Table 10 Connect No. & weight value - Case 3	44
Table 11 Connect No. & weight value - Case 4	44
Table 12 Predictive model verification	45



List of Figures

Fig. 1 Flux cored arc welding	3
Fig. 2 Classification by wire type	5
Fig. 3 FCAW wire manufacture process	5
Fig. 4 Classification of flux cored wire for steel by AWS	6
Fig. 5 Classification of flux cored wire for steel by KS	6
Fig. 6 Classification by flux cored wire	8
Fig. 7 Classification of artificial neural network	9
Fig. 8 Structure of artificial neural network	10
Fig. 9 Process calculating output value of a neuron	11
Fig. 10 Type of activation function	11
Fig. 11 Decision boundary of single layer perceptron	12
Fig. 12 Decision boundary of single layer perceptron by OR & AND Problem	12
Fig. 13 Single layer perceptron - XOR	13
Fig. 14 Multi layer perceptron - XOR	14
Fig. 15 Sequence of propagation	15
Fig. 16 Sigmoid function graph	16
Fig. 17 Univariate error function	22
Fig. 18 Weld material design flow chart	28
Fig. 19 Form of create learning data	29
Fig. 20 Sheets of training data	31
Fig. 21 Form of setting the learning model	31
Fig. 22 Data form for learning	33
Fig. 23 Network structure for prediction of weld result	34
Fig. 24 Form for learning	34
Fig. 25 Sheets of learning data	35
Fig. 26 Relation of WMANN database	36
Fig. 27 Variation of error according to learning rate - GDR	38
Fig. 28 Variation of error according to learning rate - Rprop	38

List of Figures

Fig. 29	Network structure for prediction – Case 1	40
Fig. 30	Network structure for prediction – Case 2	41
Fig. 31	Network structure for prediction – Case 3	41
Fig. 32	Network structure for prediction – Case 4	41



플럭스 코어트 와이어 설계 지원을 위한 탄력적 오류역전파 인공지능망의 적용성 검토

서 범 수

한국해양대학교 대학원
조선해양시스템공학과

초록

새로운 용접 재료의 개발은 매우 복잡한 작업이 요구된다. 와이어에 충전되는 재료는 다양한 화학 성분을 포함한다. 안정적인 용접을 위해서는 화학 성분의 구성 요소와 야금학적 반응에 의해 결정된다. 매우 복잡한 상호 작용으로 인해 용접의 프로세스를 정량적으로 분석하는 것은 거의 불가능하다. 따라서 용접 재료의 설계는 설계자의 기본적인 용접에 대한 지식과 경험을 토대로 지금까지 수행되어왔다. 용접 재료의 개발은 대개 많은 파일럿 샘플에 대해 많은 테스트와 분석을 필요로 한다. 이 연구는 용접 재료 설계 시 이러한 시험 및 분석의 양을 줄이기 위한 용접 재료의 특성을 예측하는 시스템을 목표로 한다. 본 연구에서는 플럭스 코어트 와이어의 작업성 및 용착 금속 조성을 분석하고 예측하기 위해 인공 신경망과 데이터베이스를 이용한 예측 시스템을 개발하였다. 인공 신경망은 복잡하고 비선형적인 문제를 해결하는 매우 강력한 도구이다. 그럼에도 주어진 목적을 위해 최적의 인공신경망 모델을 찾는 방법은 아직 알려지지 않았기 때문에 적절한 인공 신경망 모델을 얻는데 어려움이 있다. 따라서 많은 시행착오와 적절한 인공

신경망 모델을 얻는데 많은 시간이 필요하다. 본 연구에서는 탄력적 오류역전과 알고리즘과 데이터베이스를 결합하여 프로그램에 적용하였고 데이터베이스와 결합을 통해 인공신경망 모델을 쉽고 빠르게 생성하여 테스트할 수 있다. 탄력적 오류역전과 알고리즘은 매우 빠른 학습 알고리즘이고 기존의 일반화된 델타 규칙과 같은 역전과 알고리즘보다 수백 배 빠른 학습결과를 보여준다.

KEY WORDS: 인공신경망, 용접, 일반화된 델타 규칙, 탄력적 오류역전과 알고리즘, DB연계시스템, 예측시스템



Applicability of Resilient Back-Propagation Neural Network for support for design of flux-cored wire

Seo, Beom Su

Department of Naval Architecture and Ocean Systems Engineering
Graduate School of Korea Maritime and Ocean University



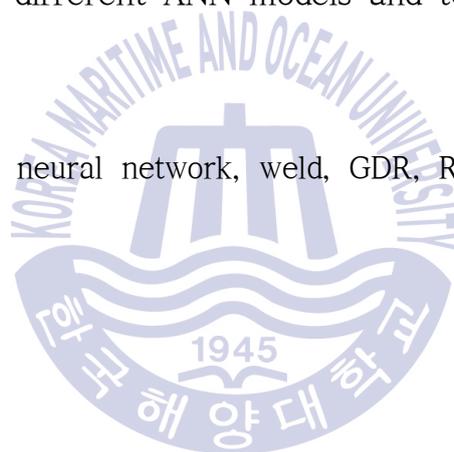
Abstract

The development of a new filler material having the required properties is a very complicated work. A filler material contains many kinds of chemical components. The properties of the weld deposited are determined by the chemical and metallurgical reaction of these components. It is nearly impossible to quantitatively analyze this process due to their highly complex interactions. Therefore the design of a filler material has been carried out up to now on the basis of fundamental metallurgical knowledge and experiences of filler material designers. The development of a filler material usually requires a lot of tests and analyses for many pilot samples. This research aims to develop the estimation system of the properties of a filler material for reducing the amount of these tests and analyses in developing a filler material. In this paper, an estimation system using an artificial neural network(ANN) and database was developed to

analyze and predict the workability and the deposited metal composition of a flux cored wire.

The neural network system is a very powerful tool to solve the complex and nonlinear problems. Nevertheless, it has a difficulty in obtaining an appropriate ANN model because the method to find optimal ANN model for any given purpose is not known yet. Therefore, it requires many trial and errors and much time to get the suitable ANN model. In this paper, the resilient backpropagation algorithm and database(DB)-coupled ANN system were applied. The resilient backpropagation algorithm is a very fast learning algorithm and shows the learning result several hundred times faster than the conventional backpropagation algorithm. The DB-coupled system can make many different ANN models and test easily and rapidly.

KEY WORDS: artificial neural network, weld, GDR, Rprop, DB-coupled system, estimation system



제 1 장 서 론

1.1 연구 배경 및 필요성

용접 품질은 여러 가지 요소들이 결합되어 결정된다. 작업자의 자세, 토치의 각도, 보호 가스의 종류, 용접 재료의 성분 등 다양한 요소들이 있다. 이 중 작업자의 자세나 토치의 각도처럼 작업자에 의해 결정되는 요소를 제외하면 용접 재료의 성분이 매우 중요한 비중을 차지한다. 다양한 용접법과 재료가 있지만 본 연구에서는 조선소에서 가장 많이 사용되는 용접법인 플럭스코어드아크용접(Flux-Cored Arc Welding, 이하 FCAW)에 사용되는 플럭스코어드와이어(Flux Co red Wire, 이하 FCW)에 대해 다룬다. FCW는 크게 와이어(Wire)와 플럭스(Flux)로 나눌 수 있고 그 비율은 공정과 재료에 따라 크게 달라지고 주로 플럭스의 성분에 의해 용접 품질이 크게 좌우된다. 플럭스(Flux)는 100가지가 넘는 화학 성분들의 조합에 의해 설계되고 통상 20~40가지의 플럭스(Flux)를 사용하여 설계한다. 이 과정에서 정해진 규칙이나 방법이 없기 때문에 설계자의 경험이나 지식이 매우 크게 작용하며 설계 경험이 부족한 설계자는 많은 시행착오를 거쳐야 상용화가 가능한 제품을 만들 수 있다. 이러한 이유로 시간과 비용이 많이 소모되며 설계자가 본인이 목적으로 하는 방향으로 설계를 했을 때 원하는 결과가 안 나온다면 딜레마(Dilemma)에 빠질 수 있다.

본 연구는 이러한 용접재료설계에 기존의 인공신경망보다 속도와 학습효과가 월등히 뛰어난 탄력적 오류역전과 신경망을 적용하여 용접재료설계의 결과에 대한 예측의 효용성을 검토하였다.

1.2 기존의 연구

용접 분야에서 인공지능은 예전부터 사용되어 왔다. 비파괴검사인 초음파 검사나 방사선 검사를 통해 얻은 데이터를 통해 슬래그 혼입(Slag inclusion), 기공(Porosity) 등의 체적결함에 대해서 용접 불량에 대해 판별을 하거나 용접 비드(Weld bead)의 형상을 촬영한 후 영상 처리를 통해 데이터를 받아 용접의 품질을 검사하는 방법도 연구됐었다. 최근에는 통계적 회귀 모형과 인공 신경망을 이용하여 용접부의 인장강도를 예측하는 연구도 있었다. 데이터가 많아지면서 일반적인 인공지능으로는 예측에 한계가 있어 요즘은 딥 러닝(Deep learning)을 이용하여 예측을 하려고 하고 있고 딥 러닝(Deep learning)에 사용되는 알고리즘들이 통계학과 관련이 많아 인공 신경망과 통계에 관련된 논문도 나오기 시작했다.

최근 4차 산업 혁명으로 머신 러닝(Machine learning), 빅 데이터(Big data) 등이 대두되고 있고 용접 분야는 변수가 매우 많아 상황별로 사람이 모든 걸 예측하기가 어렵다. 만약 머신 러닝(Machine learning)을 통해 용접 재료 설계의 예측이 가능하면 자원 및 시간을 절약이 가능하기 때문에 본 연구에서는 그 가능성을 검토해 보았다.

1.3 연구의 목적

본 연구의 목적은 기존의 경험과 추측에 의해 용접 재료를 설계하는 방법을 벗어나 머신 러닝(Machine learning)을 이용하여 설계하는 방법이다. 그 동안 쌓여있는 다량의 데이터가 있기 때문에 여러 상황별 학습이 가능하고 그 상황에 맞는 새로운 설계를 예측해 볼 수 있다. 학습에 사용되는 알고리즘은 다양한 것들이 있는데 그 중 본 연구에는 일반화된 델타 규칙(Generalized delta rule), Rprop(Resilient propagation)을 이용하여 예측 모델을 생성 후 검증을 하였다.

제 2 장 플럭스 코어드 아크 용접(FCAW: Flux Cored Arc Welding)

2.1 FCAW 원리

용접 공정은 에너지원의 종류에 따라 나눌 수 있다. 에너지원의 종류에는 전기, 화학, 집속, 기계, 고상 등이 있다. 본 연구에서는 전기를 에너지원으로 사용하는 아크 용접 중 조선소에서 가장 많이 사용되는 FCAW의 예측 및 검증에 대한 내용을 다룰 예정이다.

FCAW는 연속적으로 공급되는 플럭스가 내장된 와이어와 주로 CO₂ 가스를 사용한다. 또한 세경 와이어 속에 내장된 플럭스가 용착금속을 보호할 뿐만 아니라 효율이 높고 전자세 용접이 용이하며 넓은 용접조건 범위에서 안정된 아크를 얻을 수 있어 조선 공업의 대표적인 용접방법이다. FCAW의 용접 시 원리는 Fig. 1과 같다.

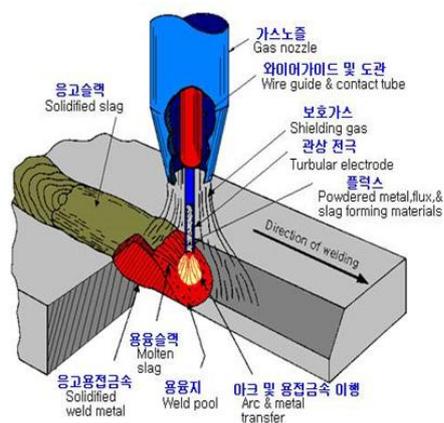


Fig. 1 Flux cored arc welding

FCAW는 조선소에서 소조, 중조, 대조의 모든 조인트(Joint)에 대하여 전자세 용접으로 사용하고 있는 용접공정이다. FCAW에서는 연속으로 공급되는 와이어 를 많이 녹이려면 와이어에서 열이 많이 발생하여야 하므로 용접기는 역극성 (DCRP, Direct Current Reverse Polarity)으로 사용하고 드물게 작은 각장을 요 구하는 경우 특별하게 정극성(DCSP, Direct Current Straight Polarity)을 사용하 는 경우가 있으며 정전압 타입으로 되어있다.

FCAW의 용접재료는 튜브형 와이어 속에 플럭스(Flux)가 채워져 있는 플럭스 코어드와이어를 사용하고 솔리드 와이어보다 전류밀도가 높아 용착속도가 크며 별도의 보호가스가 필요하다. 보호가스는 $Ar + CO_2$, $75\%Ar + 25\%CO_2$, $100\%CO_2$ 등을 사용한다. 와이어의 직경은 통상 1.2 ~ 4.8mm를 사용하고 조선소에서는 1.4mm를 가장 많이 사용한다.

용접결과에 영향을 미치는 요소는 여러 가지가 있지만 그 중 가장 큰 영향을 미치는 건 플럭스 코어드 와이어 내의 플럭스(Flux)의 성분이다. 플럭스의 성분 에 따라 작업성이나 기계적 성질에 지대한 영향을 미친다. 예를 들어 흠(Fume) 이 너무 많이 생겨 작업이 어려워지거나 아크(Arc)가 안정적으로 유지가 안 될 수 도 있고 저온에서의 충격에 강해야 한다거나 경도가 강해야 하는 등 다양한 용접 결과에 영향을 미친다. 물론 이러한 조건들을 맞추기 위해서는 다양한 작 업 조건들이 작용하지만 플럭스(Flux)가 지대한 영향을 미친다.

2.2 용접 재료

용접 재료의 구성에 있어 플럭스는 여러 가지 역할이 있다. 금속이 용해될 때 산화물의 환원이나 용융물의 표면을 피복하여 산화를 방지하기 때문에 결합 이 없는 접합이 이루어진다. 플럭스는 다양한 금속, 무기물 및 유기물 분말이 요구되는 성질을 얻을 수 있도록 적절한 비율로 혼합되어 만들어지는데 이들의 성분을 분석해보면 TiO_2 , SiO_2 , MgO 등 약 70여 가지의 화합물로 구성된다. 이 러한 화합물의 구성에 따라 작업성이나 기계적 성질, 슬래그 제거성 등이 달라

지게 된다. 따라서 플럭스의 성분을 어떻게 조합하느냐가 플럭스 코어드 와이어의 특성과 품질을 좌우하게 된다.

용접 재료 중 플럭스 코어드 와이어는 와이어 안에 플럭스가 채워져 있는 형태로써 아래의 그림과 같이 4가지 형태로 나눌 수 있다.



Fig. 2 Classification by wire type

위의 4가지 타입 중 심리스형이 와이어의 이음부가 터져서 플럭스가 새어나오는 문제가 없이 안정적인 용접 결과를 얻을 수 있어 선호되지만 플럭스를 채우는 과정이 어려워 다른 타입에 비해 고가인 단점이 있다.

플럭스 코어드 와이어의 제조공정은 금속 튜브와 튜브 내에 충전된 플럭스로 구성되어 있는데, 금속 튜브의 단면 모양이나 플럭스 조성은 매우 다양하다. 아래의 그림 Fig. 3은 FCAW의 제조공정을 개념적으로 보여주는 것인데, 먼저 10 ~ 15mm 폭을 가진 스트립을 U자형으로 성형하고, 성형된 U자형 튜브를 원하는 직경까지 인발 또는 압연하여 생산되고 있다.

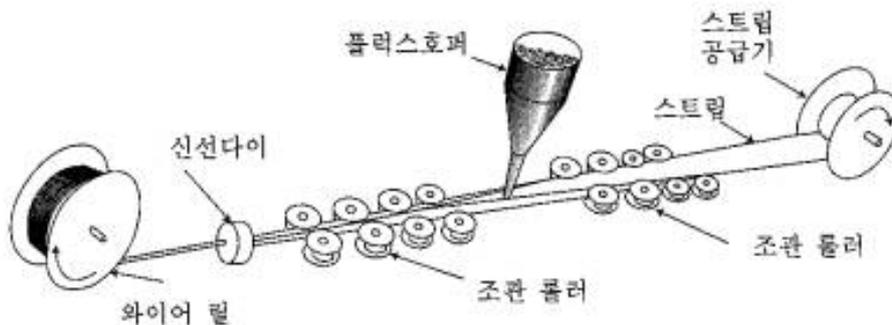


Fig. 3 FCAW wire manufacture process

플럭스 코어드 와이어의 규격은 2가지로 나눌 수 있고 그 종류는 AWS와 KW가 있으며 그 명명법은 아래의 그림 Fig .4,5와 같다.

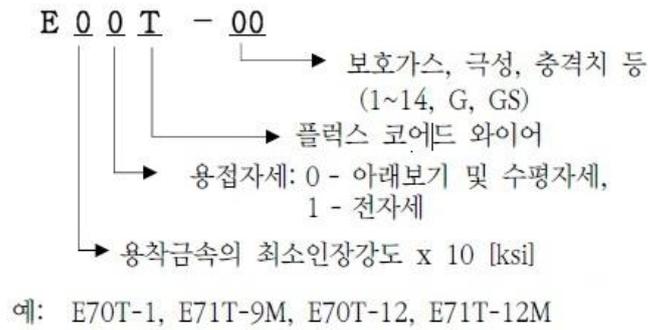


Fig. 4 Classification of flux cored wire for steel by AWS

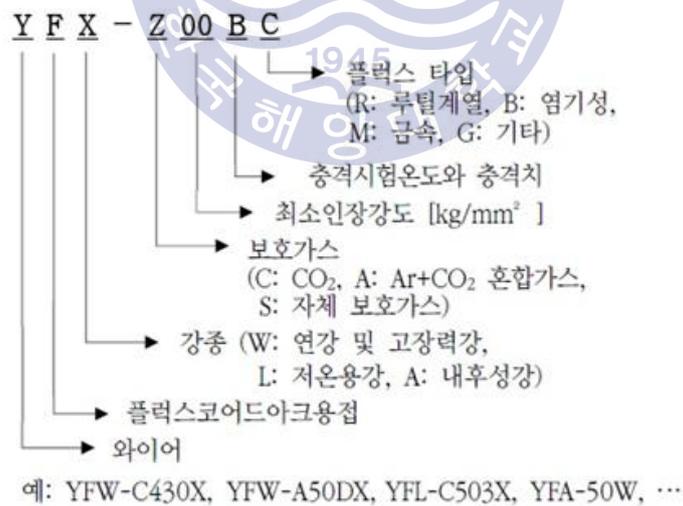


Fig. 5 Classification of flux cored wire for steel by KS

2.3 용접 결과

용접 결과는 매우 다양한 방향으로 측정과 계측이 가능하다. 그 분류를 크게 나누면 2가지로 나눌 수 있으며 하나는 용착 금속 화학 성분, 하나는 기계적 성질로 나눌 수 있다.

용착 금속 화학 성분은 용접 시 모재에 접착된 와이어를 분석한 데이터를 말하고 화학 성분비를 그 값으로 표현한다. 그 항목은 C, Si, Mn, P, S 등이 있고 주로 용착 금속은 실제 플럭스 코어드 와이어와 성분을 비교하기 위해 조사를 한다. 예를 들어 Si 성분이 10%만큼 있어야 정상이지만 그보다 적거나 많으면 정상적으로 용접이 이루어지지 않을 수 있다. 이때 용접 과정에서 이물질이 들어가거나 제작 과정에서 화합물이 정상적으로 배합이 안 되는 등 다양한 상황을 고려해봐야 한다.

반면 기계적 성질은 용착 금속 화학 성분과 다르게 여러 가지 시험을 통해 측정하게 된다. 시행하는 시험은 초기 용접 재료의 설계 목적에 따라 시험의 종류가 달라진다. 예를 들어 저온에서의 충격이 강해야 하면 영하에서의 충격치 시험을 하고 상온에서의 충격치가 중요하면 상온에서의 충격치를 시험하게 된다. 또한 이러한 시험들이 다른 시험결과에 영향을 미칠 수도 있다. 그 중 기계적 성질은 용착 금속 화학 성분에 영향을 받을 수 있다. 특히 충격치에는 Si 성분이 지대한 영향을 끼친다. 고장력강에는 일반적인 용접 재료에 비해 Si성분이 많이 들어가는 편이다. 하지만 Si가 많아지면 용접이 어려워져 전자세 용접이 불가능하게 되고 아래보기나 필릿 용접에만 사용될 수 있다.

위에서 설명한 다양한 조건에 따라 용접 재료의 종류가 달라진다. 아래의 그림은 원하는 인장 강도, 보호 가스(Self shield), 화학 성분 등에 따라 용접 재료를 구분해 놓은 것이다.

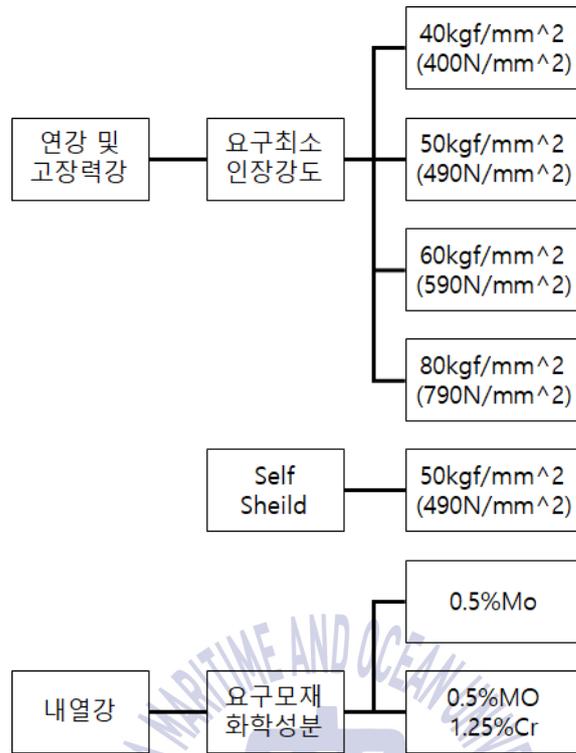


Fig. 6 Classification of flux cored wire by maker

제 3 장 인공신경망

3.1 인공신경망의 분류 및 종류

인공신경망에는 여러 가지 분류가 있고 크게 입력값이 2진수일 때와 연속적인 값일 때로 나눌 수 있고 목표 출력값이 정해져 있을 때와 정해지지 않을 때로 나눌 수 있으며 그 분류는 아래의 그림과 같이 나눌 수 있다.

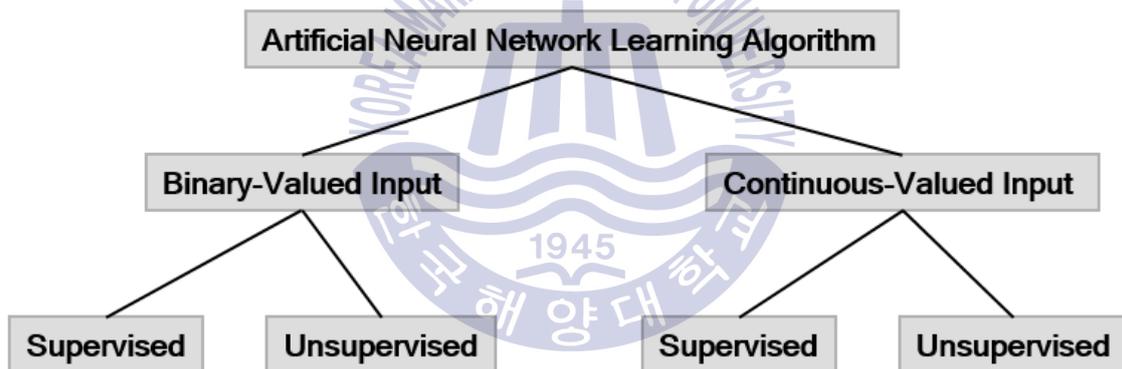


Fig. 7 Structure of artificial neural network

목표 출력값으로 분류를 나눌 때는 목표 출력값이 있을 때를 감독학습, 없을 때를 무감독학습이라 부른다. 무감독학습은 목표값 없이 학습 데이터만 입력하여 스스로 연결가중치들을 학습시키고 주로 데이터들 사이의 연관성을 찾기 어렵거나 분류를 나눠야할 때 사용된다. 감독학습은 입력이 주어짐에 따라 원하는 출력값이 활성화 되도록 가중치를 조절하는 학습을 말하고 정해진 입력값에 정해진 목표 출력값이 나오도록 학습 시키는 것을 말한다. 본 연구에서는 입력값과 목표 출력값이 정해져 있는 감독학습을 다룰 예정이다.

3.2 인공신경망의 구조

인공신경망의 구조는 입력층(Input layer), 은닉층(Hidden layer), 출력층(Output layer)으로 나눌 수 있고 각층마다 여러 개의 뉴런(Neuron)을 가지고 있으며 층간의 뉴런들 사이에는 가중치(Weight)가 있다. 바이어스(Bias)는 활성화 함수의 중심축을 좌우로 움직이는 역할을 하고 입력층이나 은닉층의 전체적인 분포에 따라 최대한 많은 값이 학습에 영향을 미치도록 조절된다. 물론 없는 경우도 있지만 대부분 출력층(Output layer)을 제외한 각층에 1개씩 배치되어 있다. 또한 모든 출력값은 활성화 함수를 통해 출력값이 나오게 된다.

우선 인공신경망의 전체적인 구조인 입력층, 은닉층, 출력층의 모습은 아래의 그림과 같다.

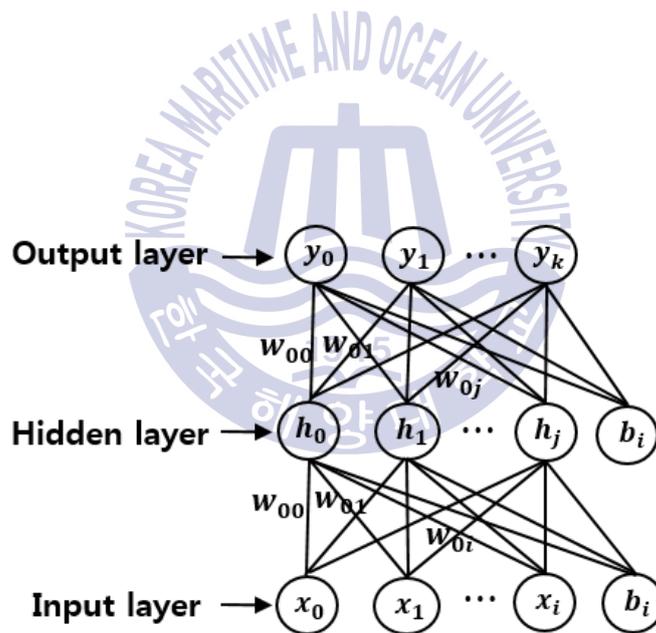


Fig. 8 Structure of artificial neural network

입력층의 뉴런(Neuron)들은 은닉층의 가중치와 결합하여 은닉층의 뉴런들로 전파가 되고 은닉층의 뉴런들도 출력층의 가중치와 결합하여 최종 출력을 하게 된다. 이때 바이어스(Bias)는 보다 정확한 출력을 위해 사용되는 변수로 작용한다.

인공신경망에서 가중치는 층과 층 사이의 모든 뉴런(Neuron) 연결에 사용되고 입력값 $x_0 \sim x_n$ 와 가중치 $w_0 \sim w_n$ 를 곱한 값을 모두 더하여 하나의 값으로 만들고 이 표현은 아래의 Fig. 9와 같다.

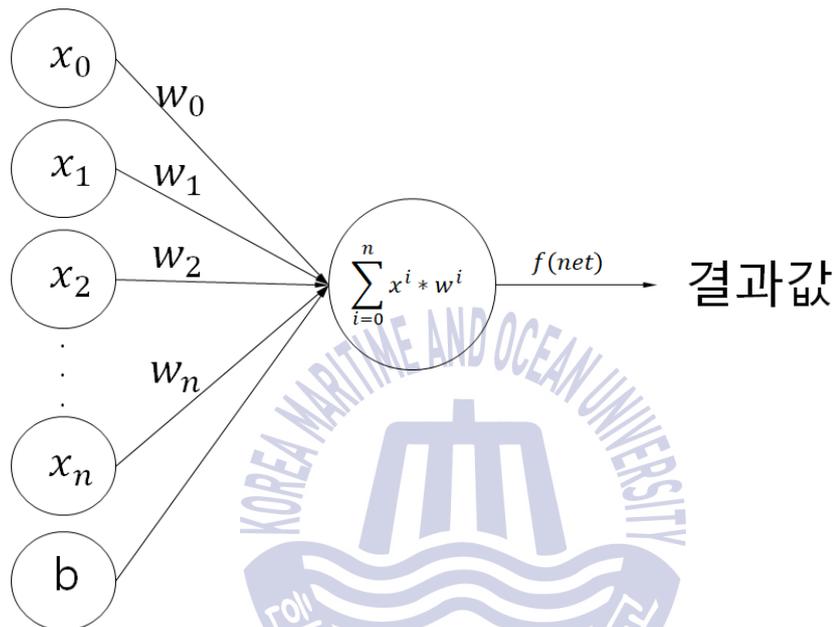


Fig. 9 Process calculating output value of a neuron

위의 Fig. 9처럼 마지막 결과값이 계산되며 이때 사용되는 함수를 활성화 함수라고 한다. 활성화 함수는 대체로 출력이 0 ~ 1사이의 값이 나오는 함수를 사용하며 그 종류는 Fig. 10과 같으며 본 연구에서 사용되는 활성화 함수는 시그모이드 함수(Sigmoid function)이다.

3.3 퍼셉트론(Perceptron)

퍼셉트론(Perceptron)은 사람 뇌의 신경계를 구성하는 주된 세포인 뉴런(Neuron)들 사이의 전기적, 화학적 신호를 주고받으며 거기서 나오는 결과 출력의 과정을 도식화 한 것이다. 여기서 층이 몇 개냐에 따라 단층, 다층으로 나누는데 입력층, 출력층만 있다면 단층, 은닉층을 포함하고 있다면 다층이 된다.

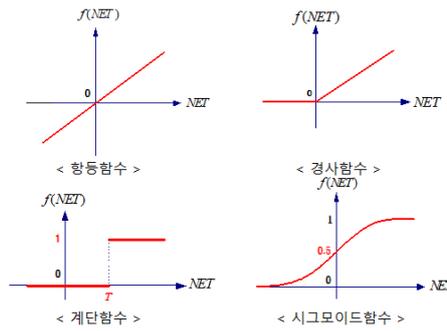


Fig. 10 Type of activation function

인공신경망을 통해 AND나 OR문제를 해결할 때 해답의 기하학적 표현을 결정경계(Decision boundary)로 표현할 수 있고 Fig. 11에 단층 퍼셉트론(Single layer perceptron)에 대한 결정경계(Decision boundary)의 예시가 나와 있다.

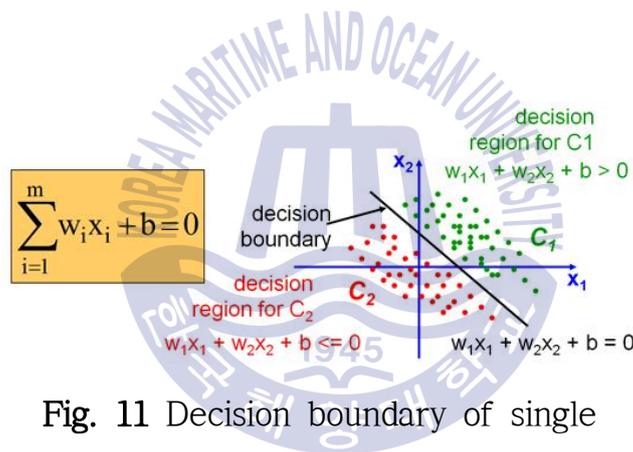


Fig. 11 Decision boundary of single layer perceptron

Fig. 11의 구역을 결정경계(Decision boundary)에 따라 C1과 C2로 나눌 수 있고 C1은 결과값이 활성화가 된 영역, C2는 결과값이 비활성화된 영역으로 구분 지을 수 있다.

Fig. 12의 빨간점이 활성화된 결과값이고 단층 퍼셉트론은 Fig. 12와 같은 AND나 OR문제 같은 선형적인 문제는 해결이 가능하다. 하지만 비선형적인 데이터의 학습이 불가능하다는 게 매우 큰 단점이다. 그 예시로 비선형 문제인 XOR문제를 예로 들 수 있다. XOR문제는 Table 1과 같은 입력값이 입력되었을 때 출력값이 활성화되어야 한다.

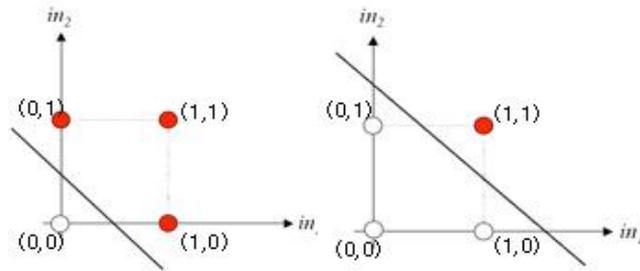


Fig. 12 Decision boundary of single layer perceptron by OR & AND Problem

Table 1 XOR problem

x_0	x_1	y
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 13의 빨간색 점들만 활성화되어야 XOR문제가 해결이 되고 Fig. 13의 결정경계는 단층 퍼셉트론(Single layer perceptron)이 XOR문제를 해결할 수 없다는 단편적인 형태를 볼 수 있다.

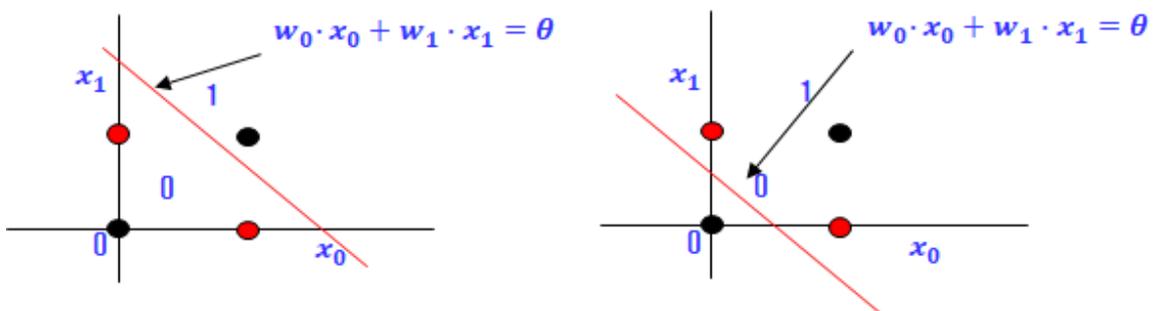


Fig. 13 Single layer perceptron - XOR

하지만 Fig. 14의 결정경계와 같이 다층 퍼셉트론(Multi-layer perceptron)은 비선형적인 문제를 해결하기 위해 은닉층을 추가하였고 다층 퍼셉트론(Multi-layer perceptron)은 Fig. 14처럼 은닉층(Hidden layer)이 추가되어 비선형적인 데이터의 학습이 가능하게 된다.

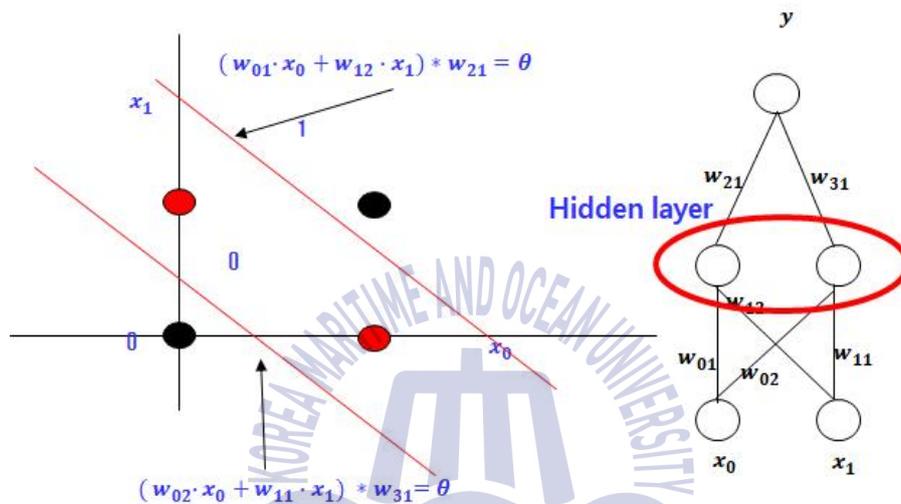


Fig. 14 Multi layer perceptron - XOR

제 4 장 학습 방법

4.1 기존의 일반화된 델타 규칙(generalized delta rule, GDR)

델타 규칙(Delta rule)은 단층 신경망을 학습시키는 방법이고 단층에서 은닉층이 추가된 다층 신경망을 일반화된 델타 규칙(Generalized delta rule)이라 부른다. 일반화된 델타 규칙(Generalized delta rule)의 특징 중 하나는 역전파(Backward propagation)이다. 역전파 알고리즘이란 인공지능의 한 분야로 경사하강법과 같은 최적화 방법과 함께 인공신경망(Artificial neural network, ANN)을 학습시키기 위한 기본적이고 일반적인 알고리즘이라고 할 수 있다. 역전파 알고리즘은 학습 벡터로부터 실제 출력값을 출력하고 목적값과 실제 출력값의 차이인 오차를 계산하여 각 층에 전달하는 전파(Propagation) 단계와 역전파된 오차를 이용하여 가중치를 수정하는 가중치 수정 단계로 구성되고 아래 Fig. 15가 그 모습을 간략하게 보여준다.

역전파 알고리즘에는 몇가지 특징이 있다. 초기 가중치는 무작위로 주어지고 그 범위는 일반적으로 -1에서 1사이의 값을 사용한다. 본 연구에서도 -1에서 1사이의 값으로 가중치 초기 설정을 하였다. 또한 각 층에 여러 노드가 있고 이 노드를 하나의 뉴런으로 가정한다. 그렇기 때문에 뉴런을 지날 때마다 활성화 함수가 적용되고 그 출력값을 Output값으로 한다. 만약 Output값이 마지막으로 출력되는 층이 있다면 그 층이 출력층이 된다. 역전파 알고리즘의 계산 순서 중 처음 입력값을 넣고 입력값에 대한 출력이 어떠한 과정으로 계산되는지 아래의 식을 보면 알 수 있다.

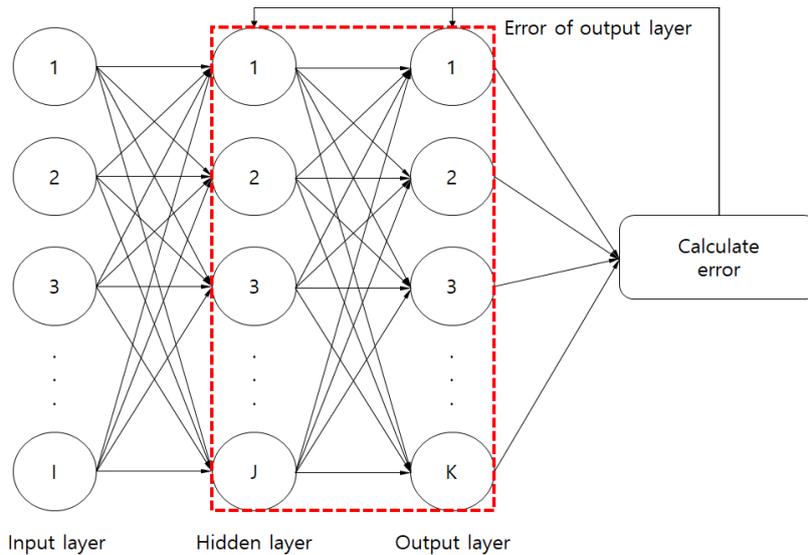


Fig. 15 Sequence of propagation

$$s_j = x_1w_{1j} + x_2w_{2j} + \dots + x_iw_{ij} \quad (1)$$

$$y_j = f(s_j) \quad (2)$$

여기서 x_i 는 입력값, w_{ij} 는 가중치, s_j 는 출력을 위한 입력값, $f(x)$ 는 활성화함수, y_j 는 출력값이다. 여기서 활성화함수는 출력값을 내기위한 함수로 일반화된 델타 규칙에서는 주로 시그모이드 함수(Sigmoid function)를 사용하고 Fig. 16에 그 식과 그래프가 있고 특징은 0에서 1사이의 값을 출력한다.

식 (1)과 (2)의 과정을 은닉층과 출력층의 모든 뉴런에 대해 계산을 하게 된다. 출력값은 시그모이드 함수로 인해 모든 값들이 0에서 1사이 값이 되므로 목표 출력값을 0과 1사이의 값으로 정규화(Normalization)시켜주어야 한다. 정규화는 목표값들 중 최대값과 최소값을 이용하여 0과 1사이의 값으로 변환할 수 있다.

정규화의 방법은 목표값이 t_1, t_2, \dots, t_n 이고 목표값 중 최대값을 t_{max} , 최소값을 t_{min} 이라고 할 때 정규화의 식은 아래와 같다.

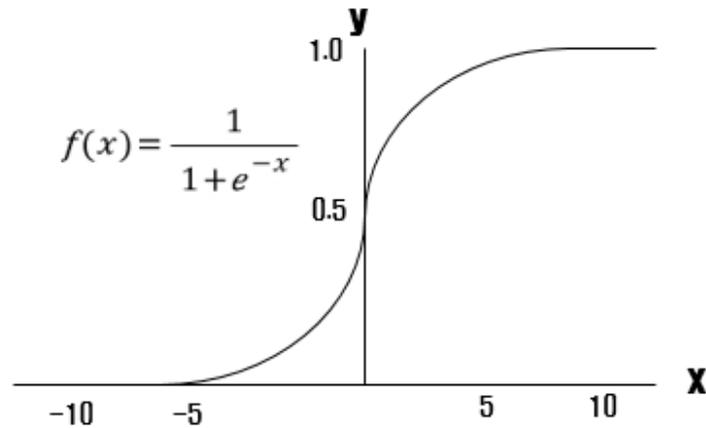


Fig. 16 Sigmoid function graph

$$t_{normal} = (t - t_{min}) / (t_{max} - t_{min}) \quad (3)$$

이 과정을 거쳐 목표값은 0과 1사이의 값이 되고 시그모이드 함수의 출력값과의 차를 이용하여 에러를 구한다. 에러값은 가중치와 바이어스값을 조정하는 지표가 된다. 에러는 목표값에서 출력값을 빼 값이고 양수와 음수가 공존하기 때문에 아래의 식(4)로 정의한다.

$$Error_{ij} = \sqrt{(t_{ij} - y_{ij})^2} \quad (4)$$

에러는 출력층 모든 뉴런에서 생성이 되고 모든 에러들을 합하여 최종 에러를 계산하고 그 식은 식(5)와 같다.

$$Error_{total} = \sum_{i=1}^n \sum_{j=1}^m Error_{ij} \quad (5)$$

출력층 뉴런마다 생성된 에러는 아래층으로 전파가 되고 은닉층과 입력층에 같은 방식으로 전달이 되며 은닉층과 입력층으로 전파되는 식은 식 (6)과 같다.

$$\Delta_{ij} = A_{ij} \times (1 - A_{ij}) \times Error_{ij} \quad (6)$$

식 (6)에서 A_{ij} 는 활성화함수에서 나온 출력값을 의미하고 $A_{ij} \times (1 - A_{ij})$ 는 활성화 함수인 시그모이드 함수의 특성에 의해 식 (7)처럼 전개된다.

$$f(x) = \frac{1}{1 + e^{-x}}, f'(x) = f(x) \times (1 - f(x)) \quad (7)$$

식 (6)에서 Δ_{ij} 는 각 뉴런의 변화량과 $Error_{ij}$ 의 곱을 나타내고 Δ_{ij} 는 은닉층과 입력층의 에러 초기화와 가중치, 바이어스의 갱신에 사용된다. 우선 은닉층과 입력층의 에러 초기화는 식(8)과 같다.

$$Error_{i-1j} = \Delta_{ij} \times w_{ij} \quad (8)$$

오류 역전파 알고리즘의 가중치 변화는 (입력층 - 은닉층), (은닉층 - 출력층) 이렇게 두 가지로 나눌 수 있다. 먼저 (입력층 - 은닉층)사이의 가중치 변화는 아래의 식 (9)과 같다.

$$w_{ij}(t+1) = w_{ij}(t) - \eta \frac{\partial E}{\partial w_{ij}(t)} \quad (9)$$

오차는 (목표값 - 출력값)이므로 가중치로 미분이 불가능하다. 따라서 chain rule을 적용하여 식 (10)과 같이 전개한다.

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial f(s_j)} \frac{\partial f(s_j)}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}} \quad (10)$$

식 (10)의 첫 번째 요소는 식(11)과 같고

$$\frac{\partial E}{\partial f(s_j)} = \sum_{k=1}^K \frac{\partial E}{\partial f(s_k)} \frac{\partial f(s_k)}{\partial s_k} \frac{\partial s_k}{\partial f(s_j)} \quad (11)$$

식 (11)의 3가지 요소는 아래와 같다.

$$\begin{aligned} \frac{\partial E}{\partial f(s_k)} &= \partial \frac{1}{2} \frac{(f(s_1) - t_1)^2 + (f(s_2) - t_2)^2 + \dots + (f(s_k) - t_k)^2}{\partial f(s_k)} \\ &= \frac{\partial \frac{1}{2} (f(s_k) - t_k)^2}{\partial f(s_k)} = f(s_k) - t_k \end{aligned} \quad (12)$$

$$\frac{\partial f(s_k)}{\partial s_k} = f'(s_k) = (1 - f(s_k))f(s_k) \quad (13)$$

$$\begin{aligned} \frac{\partial s_k}{\partial f(s_j)} &= \frac{\partial (f(s_1)w_{1k} + f(s_2)w_{2k} + \dots + f(s_j)w_{jk})}{\partial f(s_j)} \\ &= \frac{\partial f(s_j)w_{jk}}{\partial f(s_j)} = w_{jk} \end{aligned} \quad (14)$$

따라서 식 (11)은 아래 식 (15)과 같이 정의되고

$$\therefore \frac{\partial E}{\partial f(s_j)} = \sum_{k=1}^K (f(s_k) - t_k)(1 - f(s_k))f(s_k)w_{jk} \quad (15)$$

식 (10)의 두 번째, 세 번째 요소는 아래 식 (16), (17)과 같다.

$$\frac{\partial f(s_j)}{\partial s_j} = f'(s_j) = (1 - f(s_j))f(s_j) \quad (16)$$

$$\frac{\partial s_j}{\partial w_{ij}} = \frac{\partial (x_1w_{1j} + x_2w_{2j} + \dots + x_iw_{ij})}{\partial w_{ij}} = \frac{\partial x_iw_{ij}}{\partial w_{ij}} = w_i \quad (17)$$

위의 식들을 종합하면 식 (10)는 식 (18)과 같이 정리된다.

$$\therefore \frac{\partial E}{\partial w_{ij}} = \left\{ \sum_{k=1}^K (f(s_k) - t_k)(1 - f(s_k))f(s_k)w_{jk} \right\} (1 - f(s_j))f(s_j)x_i \quad (18)$$

(은닉층 - 출력층)사이의 가중치는 (입력층 - 은닉층)과 비슷하고 조금 더 간단하다. 가중치 변화의 식은 식 (19), (20)과 같다.

$$w_{jk}(t+1) = w_{jk}(t) - \eta \frac{\partial E}{\partial w_{jk}(t)} \quad (19)$$

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial f(s_k)} \frac{\partial f(s_k)}{\partial s_k} \frac{\partial s_k}{\partial w_{jk}} \quad (20)$$

식 (20)의 세 요소는 식 (10)의 세 요소와 비슷하고 정리하면 아래의 식과 같

이 정리된다.

$$\begin{aligned}\frac{\partial E}{\partial f(s_k)} &= \partial \frac{1}{2} \frac{(f(s_1) - t_1)^2 + (f(s_2) - t_2)^2 + \dots + (f(s_k) - t_k)^2}{\partial f(s_k)} \\ &= \frac{\partial \frac{1}{2} (f(s_k) - t_k)^2}{\partial f(s_k)} = f(s_k) - t_k\end{aligned}\quad (21)$$

$$\frac{\partial f(s_k)}{\partial s_k} = f'(s_k) = (1 - f(s_k))f(s_k)\quad (22)$$

$$\begin{aligned}\frac{\partial s_k}{\partial w_{jk}} &= \frac{\partial (f(s_1)w_{1k} + f(s_2)w_{2k} + \dots + f(s_j)w_{jk})}{\partial w_{jk}} \\ &= \frac{\partial f(s_j)w_{jk}}{\partial w_{jk}} = f(s_j)\end{aligned}\quad (23)$$

$$\therefore \frac{\partial E}{\partial w_{jk}} = (f(s_k) - t_k)(1 - f(s_k))f(s_k)f(s_j)\quad (24)$$

위의 과정들을 통해 은닉층과 입력층의 $Error_{ij}$ 와 Δ_{ij} 가 갱신된 뒤 후 w_{ij} 와 b_{ij} 를 목표값이 나오도록 재설정한다. w_{ij} 와 b_{ij} 의 재설정 식은 아래와 같다.

$$\begin{aligned}b_{ij(New)} &= b_{ij(old)} + l \times \Delta_{ij} + m \times \Delta b_{ij} \\ \Delta b_{ij} &= b_{ij(New)} - b_{ij(old)}\end{aligned}\quad (25)$$

$$\begin{aligned}w_{ij(New)} &= w_{ij(old)} + l \times \Delta_{ij} + m \times \Delta w_{ij} \\ \Delta w_{ij} &= w_{ij(New)} - w_{ij(old)}\end{aligned}\quad (26)$$

여기서 l 은 학습률(Learning rate), m 은 모멘텀(Momentum)이다. w_{ij} 와 b_{ij} 가 모두 재설정이 되면 다시 입력값을 넣어 재계산을 한다. 이 과정을 무수히

많이 반복하여 식 (5)의 오차를 줄여 입력값에 대한 예측값을 얻어 내는 게 목적이다.

4.2 탄력적 오류역전과 알고리즘

Rprop(탄력적 오류역전과 알고리즘)은 resilient back-propagation의 약자로 지역 최대 에러 함수(Local minimum of the error function)를 찾기 위해 신경망 가중치를 수정한 전통적인 역전파 알고리즘에 기반한다. 그러므로 에러 함수($\partial E/\partial w$)의 기울기는 근을 찾기 위한 가중치에 대해 계산한다. 특히, 가중치는 지역 최소값에 다달을 때까지 편미분(Partial derivatives)의 반대 방향으로 진행 하도록 수정했다. 이 기본 개념은 단변량(Univariate) 에러 함수 변화를 통해 알 수 있고 그 모습을 아래의 Fig. 16과 같이 계략적으로 설명 할 수 있다.

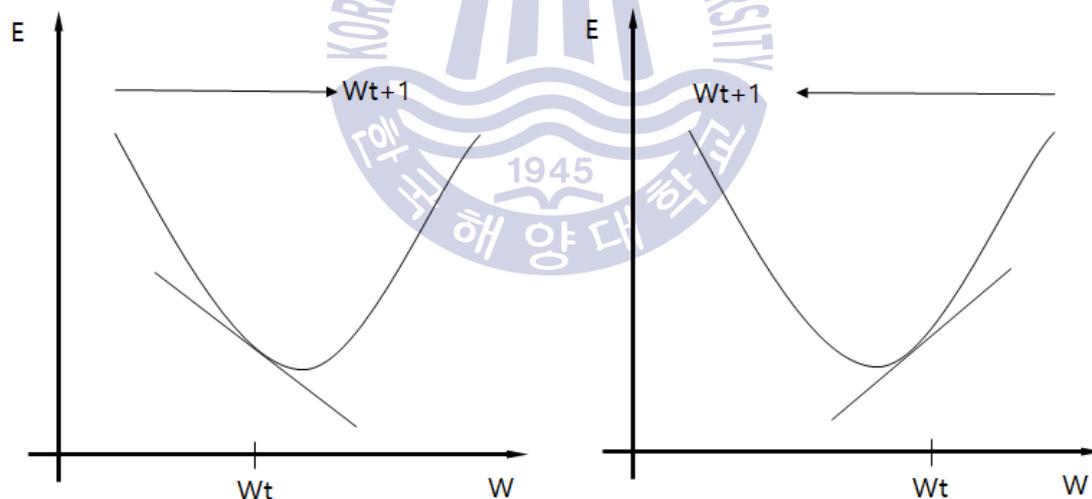


Fig. 17 Univariate error function

Fig. 17은 가중치의 변화에 따른 에러값을 표현한 그림이다. 이 그림을 통해 알 수 있는 건 편미분이 음수이면 가중치는 증가하고 편미분이 양수이면 가중치는 감소된다. 모든 편미분은 신경망에 계산되는 함수를 기본적으로 통합 그리고 활성화 함수이므로 Chain rule을 이용하여 계산한다.

앞서 설명한 일반화된 델타 규칙과의 차이점은 기존의 역전파 알고리즘은 출력값의 변화량과 에러를 이용하여 가중치와 바이어스를 갱신했었고 탄력적 오류역전파 알고리즘은 가중치에 대한 오차의 변화율과 변화율의 부호 변화를 이용하여 갱신을 한다.

또한 갱신의 방법을 개선하여 탄력적 오류역전파 알고리즘의 종류는 총 4가지로 나눌 수 있고 Rprop+, Rprop-, iRprop+, iRprop- 순서로 개선되었다. 개선 과정은 Rprop+가 가장 먼저 발표되었고 Rprop+를 간략화한 것으로 시킨 것이 Rprop- 이다. 간략화된 과정은 가중치 갱신 과정에서 가중치에 대한 오차 변화율이 같거나 다를 때 혹은 '0'일 때 3가지 경우로 나누어 모든 경우의 수마다 가중치 갱신식을 다르게 설정하였고 가중치에 대한 전 단계와 현 단계의 오차 변화율의 부호가 같을 때는 식 (27), (28), (29)와 같고

$$\Delta_{ij}^{(t)} := \min(\Delta_{ij}^{(t-1)}, \eta^+, \Delta_{\max}) \quad (27)$$

$$\Delta w_{ij}^{(t)} := -\text{sign}\left(\frac{\partial E}{\partial w_{ij}}\right)^{(t)} \cdot \Delta_{ij}^{(t)} \quad (28)$$

$$w_{ij}^{(t+1)} := w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \quad (29)$$

전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호가 다를 때는 식 (30), (31), (32)와 같으며

$$\Delta_{ij}^{(t)} := \max(\Delta_{ij}^{(t-1)}, \eta^-, \Delta_{\min}) \quad (27)$$

$$w_{ij}^{(t+1)} := w_{ij}^{(t)} - \Delta w_{ij}^{(t-1)} \quad (28)$$

$$\left(\frac{\partial E}{\partial w_{ij}}\right)^{(t)} := 0 \quad (29)$$

가중치에 대한 오차 변화율이 '0'일 때는 식 (30), (31) 와 같다.

$$\Delta w_{ij}^{(t)} := -\text{sign}\left(\frac{\partial E}{\partial w_{ij}}\right)^{(t)} \cdot \Delta_{ij}^{(t)} \quad (30)$$

$$w_{ij}^{(t+1)} := w_{ij}^{(t)} + \Delta w_{ij}^{(t)} \quad (31)$$

Rprop-는 이를 간략화시켜 각각의 경우에 상관없이 부호 판별을 통해 가중치를 갱신시켰고 전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호가 같을 때는 식 (32)와 같고

$$\Delta_{ij}^{(t)} := \min(\Delta_{ij}^{(t-1)} \cdot \eta^+, \Delta_{\max}) \quad (32)$$

전 단계와 현 단계의 가중치에 대한 오차 변화율이 다를 때는 식 (33)과 같으며

$$\Delta_{ij}^{(t)} := \max(\Delta_{ij}^{(t-1)} \cdot \eta^-, \Delta_{\min}) \quad (33)$$

전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호 변화와 상관없이 공통적으로 식 (34)를 적용시켰다.

$$w_{ij}^{(t+1)} := w_{ij}^{(t)} - \text{sign}\left(\frac{\partial E}{\partial w_{ij}}\right)^{(t)} \cdot \Delta_{ij}^{(t)} \quad (34)$$

다시 정리하면 공통된 식을 가중치에 대한 오차 변화율의 부호 변화에 따라 가중치 갱신을 개별적으로 처리하던 부분을 마지막에 공통적으로 적용되도록 간소화 시킨 것이고 식 (34)의 Sign은 가중치에 대한 오차변화율의 부호에 따라 + 혹은 -로 바뀌게 되고 가중치에 대한 오차변화율의 부호가 + 이면 가중치가 증가할 때 오차값도 증가하므로 가중치를 낮춰야하기 때문에 Sign의 부호는 +가 되어야 하고 가중치에 대한 오차변화율의 부호가 - 이면 가중치가 증가할 때 오차값이 감소하므로 가중치를 높여야하기 때문에 Sign의 부호는 -가 된다.

또한 Rprop+에서 iRprop+로 변하는 개선 과정에서는 전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호가 변하고 오차값이 이전 오차값보다 증가할 때 식 (35)를 사용하도록 개선되었고 전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호가 변하더라도 오차값이 커지면 변화를 최소화시키기 위함이고 오차값이 이전보다 작아지면 가중치 갱신에 이전 가중치 변화량을 사용하지 않게 하여 오차를 더 탄력적으로 줄일 수 있도록 하였다.

$$w_{ij}^{(t+1)} := w_{ij}^{(t)} - \Delta w_{ij}^{(t-1)} \quad (35)$$

Rprop-에서 iRprop-로 변하는 개선 과정에서는 전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호가 변할 때 오차 변화율을 '0'으로 초기화하도록 식(36)을 추가하였고 전 단계와 현 단계의 가중치에 대한 오차 변화율의 부호가 변하면 오차값을 그래프로 표현하였을 때 변곡점을 지난 것 이므로 가중치의 변화를 최소화시키기 위함이다.

$$\left(\frac{\partial E}{\partial w_{ij}} \right)^{(t)} = 0 \quad (36)$$

본 연구에서는 가장 최근에 개선된 iRprop-를 사용하였고 가중치 갱신 과정에 있어 가중치 변화량을 조절하여 최적의 조절값을 찾기 위해 식(37)과 같이 가중치 변화량에 학습률을 곱하였다.

$$w_{ij}^{(t+1)} := w_{ij}^{(t)} - \eta \cdot \text{sign} \left(\frac{\partial E}{\partial w_{ij}} \right)^{(t)} \cdot \Delta_{ij}^{(t)} \quad (37)$$

식 (27)부터 식 (37)까지 사용된 변수 중 $\Delta_{ij}^{(t)}$ 는 현 단계의 가중치 조절량, $\Delta_{ij}^{(t-1)}$ 는 전 단계의 가중치 조절량이고 η 는 학습률이다. 만약 가중치에 대한 오차 변화율의 부호가 이전과 같다면 η^+ 를 이용하여 가중치의 조절량을 증가시키고 가중치에 대한 오차 변화율의 부호가 이전과 다르다면 η^- 를 이용하여 가중치의 조절량을 감소시킨다. 여기서 η^+ , η^- 는 $0 < \eta^- < 1 < \eta^+$ 의 값을 갖고 본 연구에서 η^+ 는 1.2, η^- 는 0.5를 사용하였다. 가중치 갱신 과정에서 가중치가 무한히 작아지거나 커지지 않도록 가중치 조절량의 최대값 Δ_{\max} 와 최소값 Δ_{\min} 을 설정하여 두 값의 사이 범위가 되도록 조정한다. 본 연구에서 Δ_{\max} 는 50, Δ_{\min} 은 $1e^{-6}$ 으로 설정하였다. 변화된 과정들은 오차 변화를 그래프로 표현할 때 변곡점을 지나는 과정을 고려한 것이고 변곡점의 의미는 오차의 최소값이 될 수 있다.

위의 설명들은 탄력적 오류역전과 알고리즘의 특징들만 설명을 하였고 활성

함수나 출력값을 얻는 과정은 일반화된 델타 규칙과 동일하다. 하지만 위와 같은 특징이 오차를 줄여나가는 데 지대한 영향을 미치고 시간을 단축시킬 수 있다.



제 5 장 인공지능망을 이용한 용접 재료 설계 지원

5.1 용접 재료 설계 과정

용접 재료 설계는 설계를 시작하기에 앞서 설계하고자 하는 용접 공정, 설계 목적 등을 결정하고 설계를 진행하게 된다.

용접 재료 설계 시 한 번에 여러 개의 용접 재료를 설계하고 기존의 설계 데이터와 비교하면서 설계 목적에 맞게 설계한다. 이 과정에서 Flux 성분의 비율을 조정함에 따라 화학 성분도 동시에 바뀌게 되고 화학성분 약간의 변화도 용접 결과에 지대한 영향을 미치고 이러한 이유로 설계자는 Flux 변화에 따른 화학 성분의 변화를 확인하며 설계한다.

용접 재료 설계가 완료되면 해당 용접 재료 설계에 대해서 실제 용접 테스트를 실행한다. 용접 테스트는 직접 용접을 한 후 앞서 말했던 기계적 성질과 용착 금속 성분 등을 조사한다. 기계적 성질을 측정하는 시험에는 충격 시험인 Izod, Charpy 등이 있고 인장력을 시험하는 Tensile 시험이 있다. 이러한 시험이 한 번에 끝날 수도 있지만 상황에 따라 여러 번에 걸쳐 테스트를 시행할 수도 있다. 그렇기 때문에 한 설계에 여러 개의 테스트 결과가 존재할 수 있다.

이러한 테스트 결과를 확인한 후 초기의 설계 목적과 비교하여 초기의 설계 목적에 어긋나는 테스트 결과가 나오게 되면 Flux 성분을 다시 조정하는 재설계 과정을 거치게 된다. 하지만 테스트 결과를 한번 얻어내기 위해서는 상기한 다양한 시험을 거쳐야 하므로 많은 시일이 소요되고 그 과정에서 적지 않은 비용이 소요된다. 이러한 일련의 과정을 Fig. 18의 Flow chart로 나타낼 수 있다.

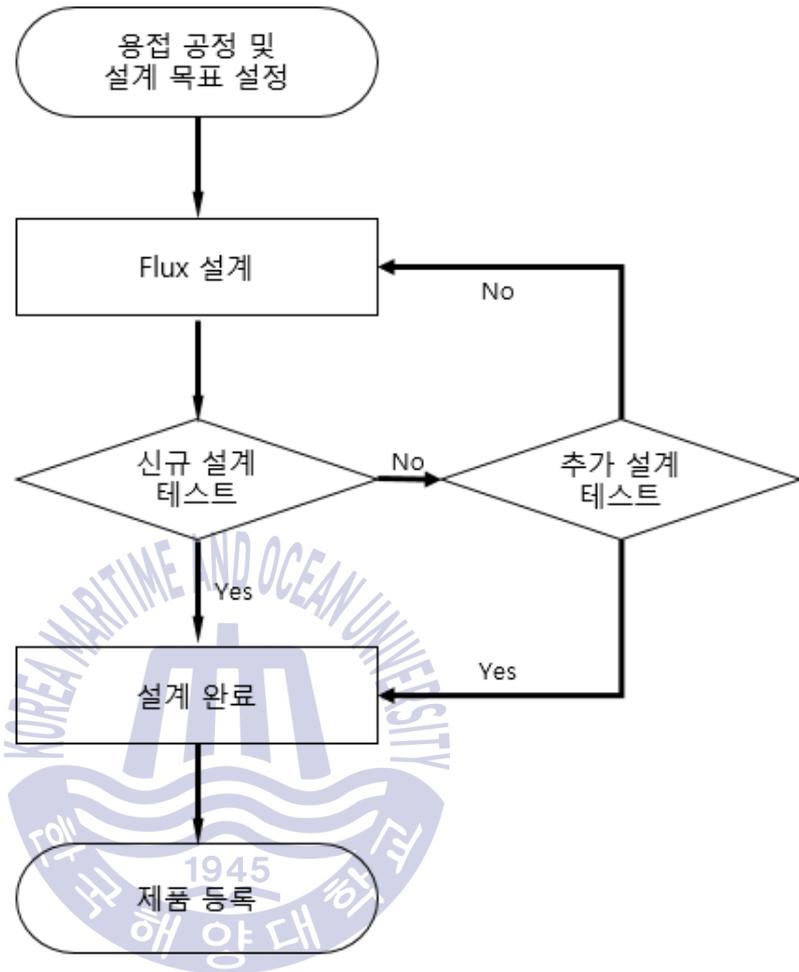


Fig. 18 Weld material design flow chart

5.2 용접 재료 설계 지원 인공지능망 시스템

용접 재료 설계를 지원하기 위한 인공지능망 시스템은 Weld Metal Artificial Neural Network의 약자로 WMANN으로 명명하기로 한다.

학습 데이터를 만들기 위해서는 입력값과 목표값을 어떤 항목으로 할지 먼저 설정해야한다. 입력값은 Flux 비율, 화학 성분, 테스트 결과 등 여러 가지가 될

수 있다. 하지만 목표값은 우리가 추정해야 할 항목을 선택해야 하기 때문에 테스트 결과 이외에는 대부분 입력하거나 계산되어 나오는 값들이기 때문에 주로 테스트 결과를 목표값으로 설정한다. 테스트 결과는 용착 금속 성분, 온도에 따른 충격치 등 다양한 분류들이 있기 때문에 어떤 분류를 선택함에 따라 다양한 상관관계를 추정해 볼 수 있다. 또한 테스트 결과안의 항목으로 입력값과 목표값을 설정 할 수도 있다.

앞서 설명한 데이터베이스와 학습 데이터 항목들을 이용하여 학습 데이터를 생성할 수 있다. 하지만 학습 데이터를 생성할 때 주의해야 할 부분이 있다. 입력값의 항목 중 값이 없는 것은 '0'으로 대체할 수 있다. 이유는 어떤 값은 있고 또 다른 값이 없다고 할 때 없는 값이 목표값에 영향을 미칠 수 있기 때문이다. 하지만 목표값의 항목이 '0'인 것과 값이 없는 것에 대한 건 구분이 필요하다. 값이 '0'인건 항목에 값이 존재하기 때문에 데이터 생성이 포함을 시켜야 한다. 하지만 값이 없는 것을 항목에 포함을 시켜 그 값이 '0'으로 인식이 되는 건 고려를 해 봐야 한다. 예를 들어 상온에서의 충격치를 테스트를 시행하지 않고 저온에서의 충격치 테스트만 진행하여 저온에서의 충격치만 값이 있다고 할 때 상온에서의 충격치가 '0'이라고 할 수 없다. 그렇기 때문에 데이터 생성 시 데이터 하나하나를 확인하기 어렵기 때문에 Fig. 19에서 앞서 언급한 부분을 처리하고 저장할 수 있도록 만들었다.

항목	Pi-1	Pi-2	HO-070	HO-071	HO-068
1 SiO2	12.0131	8.9879	6.6383	6.638	
2 MgO	0.7824	0.5538	0.4669	0.4669	
3 MgO	0	0.0059	0.0024	0.0023	
4 Al2O3	0.1639	0.4058	0.0524	0.0524	
5 ZrO2	0.09	1.6032	0.0587	0.0562	
6 K2O	0.1128	0.1611	0.0879	0.0872	
7 CaO	0	0.0061	0.0079	0.0078	
8 Na2O	0.0001	0	0.0142	0.0142	
9 MnO	0	0.4408	0	0	
10 MnO2	0	0	0	0	
11 Li2O	0.0393	0	0	0	
13 T_Fe	5.483	0.3843	3.6991	3.6373	
14 Fe2O3	0.0443	0.0337	0.0342	0.0313	
16 Bi2O3	0	0.0777	0	0	
25 CO2	0	0	0	0	
26 Na	0	0	0	0	
27 P2O5	0.0154	0.0072	0.0033	0.0035	
28 F	0.0537	0	0.0232	0.0232	
29 NaF	0.2056	0.257	0.1631	0.1631	
30 CaF2	0	0	0	0	
32 LiF	0	0	0	0	
33 MgF2	0	0	0	0	
37 Na2CO3	0.0001	0.0001	0.0001	0.0001	
39 Li2CO3	0	0	0	0	
41 H2O	0.0002	0.0003	0.0001	0.0001	
42 IL	0.0042	0.0053	0.0212	0.0181	
43 Ni	0.0023	3.2467	0.0019	0.0019	
44 C	0.0129	0.0035	0.0106	0.011	
45 Mo	0.0002	0	0.0001	0.0001	
46 W	0	0	0	0	
47 V	0.0044	0	0.0007	0.0007	
48 Cu	0.0007	0	0.0007	0.0007	
50 Co	0.0005	0.0027	0.0002	0.0002	
51 B	0	0	0	0	
53 P	0.0067	0.0032	0.0045	0.0046	
54 S	0.0011	0.002	0.001	0.0009	

Fig. 19 Form of create learning data

앞서 설명한 과정을 거쳐 입력값과 목표값의 항목을 결정하고 데이터를 추출해낸 후 학습을 시키기 위해 정규화 과정을 거쳐야 한다. 정규화를 하는 이유는 학습을 시킬 때 항목간의 값 차이가 너무 많이 나면 학습률을 재설정이나 정상적으로 되지 않을 수 있다. 만약 한 항목 중 하나는 값이 '20'이고 다른 값이 '0.1'이라고 할 때 같은 가중치로 계산을 하게 되면 편차가 너무 심하기 때문에 정규화를 통해 그 편차를 줄일 필요가 있고 활성화함수로 시그모이드 함수를 사용하기 때문에 결과값이 0에서 1 사이의 값이 나오기 때문에 학습을 시킬 때부터 0에서 1사이의 값으로 정규화 후 학습을 시키면 더 정확한 학습이 가능할 것이라 예상하여 본 논문에서는 입력값과 목표값을 정규화 시킨 후 학습을 시켰고 정규화 식은 식 (37)과 같다.

$$x_{normal} = (x - x_{min}) / (x_{max} - x_{min}) \quad (37)$$

x_{min} 과 x_{max} 는 한 항목의 최소값과 최고값이고 이 값들을 저장한 후 추후에 추정값의 실제값을 구할 때 사용하고 구하는 식은 식 (38)과 같다.

$$x = x_{normal} \times (x_{max} - x_{min}) + x_{min} \quad (38)$$

앞서 설명한 내용의 데이터들은 총 5개의 시트로 들어간다. 하나는 용접공정, 분류, 입력값의 항목, 출력값의 항목이 들어가 있고 나머지 4개는 원입력, 출력 데이터와 그 데이터를 정규화시킨 값과 최소값, 최대값이 저장되고 Fig. 20과 같이 생성된다.

	A	B	C	D	E	F	G	H	I		
1	용접공정	분류1	입력항목그룹	출력항목그룹	END						
2	FCAW	산성계	용접결과	용접결과							
3											
4	1	DesignNo	WM_C	WM_Si	WM_Mn	WM_P	WM_S	WM_Ti	WM_B		
5	26	CH-34									
6	27	CH-35	1	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
7	28	CH-36	2	DesignNo	WM_C	WM_Si	WM_Mn	WM_P	WM_S	WM_Ti	WM_B
8	29	CH-37	3	DesignNo	WM_C	WM_Si	WM_Mn	WM_P	WM_S	WM_Ti	WM_B
9	30	CH-38	4	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
10	31	CH-39	5	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
11	32	CH-4	6	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
12	33	CH-40	7	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
13	34	CH-41	8	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
14	35	CH-42	9	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
15	36	CH-43	10	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
16	37	CH-44	11	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
17	38	CH-45	12	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
18	39	CH-7	13	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
19	40	CH-8	14	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
20	41	HN-49	15	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
21	42	HN-50	16	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
22	43	HN-50	17	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
23	44	HN-50	18	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
24	45	HN-50	19	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
25	46	HN-50	20	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
26	47	HN-50	21	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
27	48	HN-50	22	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
28	49	HN-50	23	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
29	50	HN-50	24	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
30	51	HN-50	25	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
31	52	HN-50	26	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
32	53	HN-50	27	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
33	54	HN-50	28	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
34	55	HN-50	29	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
35	56	HN-50	30	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
36	57	HN-50	31	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
37	58	HN-50	32	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
38	59	HN-50	33	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
39	60	HN-50	34	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
40	61	HN-50	35	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
41	62	HN-50	36	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
42	63	HN-50	37	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
43	64	HN-50	38	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
44	65	HN-50	39	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
45	66	HN-50	40	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
46	67	HN-50	41	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
47	68	HN-50	42	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
48	69	HN-50	43	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
49	70	HN-50	44	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
50	71	HN-50	45	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
51	72	HN-50	46	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
52	73	HN-50	47	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
53	74	HN-50	48	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
54	75	HN-50	49	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V
55	76	HN-50	50	DesignNo	X_Factor	J_Factor	YP	TS	YR비	longatio	m20°C_V

Fig. 20 Sheets of training data

학습 데이터가 생성되고 학습을 시키기 위해선 몇 가지 설정을 해주어야 하고 그 설정은 Fig. 21과 같다.

The screenshot shows a configuration window for a learning model. It is divided into several sections:

- 모델 기본 정보 (Model Basic Info):**
 - 모델명 (Model Name): 1-Rprop-N-0%-FL
 - 분류1 (Classification 1): 산성계
 - 입력항목그룹 (Input Item Group): 용접결과
 - 출력항목그룹 (Output Item Group): 용접결과
 - 모델설명 (Model Description):
 - 모델역파일명 (Model Reverse File Name): 1. 논문용 - 48 - 학습.xls
- 모델 상세 정보 (Model Detailed Info):**
 - 학습률 (Learning Rate): 47
 - 중수 (Weight): 3
 - 입력층뉴런수 (Input Layer Neurons): 15
 - 은닉1층뉴런수 (Hidden 1 Layer Neurons): 15
 - 출력층뉴런수 (Output Layer Neurons): 8
 - 은닉2층뉴런수 (Hidden 2 Layer Neurons):
- 학습규칙 (Learning Rule):**
 - 일반화된 델타 규칙 (Generalized Delta Rule)
 - Fast learning
 - Accord Fast learning
 - Accord Deep learning

Fig. 21 Form of setting the learning model

우선 해당 학습 모델이 어떤 목적으로 학습을 시키는지 어떤 과정을 통해 학습을 하게 되는지 적어야 한다. 그 후 은닉층의 뉴런 개수를 지정하고 어떤 학습 규칙으로 학습을 시킬지 정해야 한다. 대부분 은닉층의 뉴런 개수는 입력층과 같게 하거나 그 개수의 1.2배정도로 설정한다. 그리고 학습 규칙을 어떤 걸 선택할지가 매우 중요하다. 이유는 어떤 규칙을 선택 하나에 따라 학습 횟수와 학습률 및 모멘텀이 달라지기 때문이다. 본 연구에서는 일반화된 델타 규칙과 탄력적인 오류역전파 알고리즘 두 가지만 사용하는데 아래의 표에 두 학습 규칙의 성향이 나와 있다.

Table 2 Characteristics of learning rule

	일반화된 델타 규칙	탄력적인 오류역전과 알고리즘
학습 횟수	500,000회 이상	100,000회 이하
학습률 범위	0.3 ~ 0.1	0.1 ~ 0.01
모멘텀 유무	O	X

Table 2의 데이터 중 학습 횟수가 눈에 띈 것이다. 일반화된 델타 규칙에 비해 탄력적인 오류역전과 알고리즘의 횟수가 현저하게 적다. 이유는 탄력적인 오류역전과 알고리즘의 가중에 따른 오차 변화율이 빠르게 수렴하게 만들기 때문이다. 일반화된 델타 규칙은 변화율과 상관없이 일관되게 수렴지점을 찾는 반면에 탄력적인 오류역전과 알고리즘은 변화율로 모멘텀을 대신하기 때문에 훨씬 더 빠르게 수렴치는 찾을 수 있다.

학습률의 범위는 학습 규칙마다의 계산 특성에 의해 정해진다. 우선 일반화된 델타 규칙은 너무 낮은값으로 학습률을 선택하게 되면 학습 횟수가 그만큼 대폭 늘어나게 된다. 물론 학습률을 낮추면 수렴될 때의 오차가 더 줄어들 확률이 있다. 하지만 그 차이에 비해 학습 횟수가 몇 배를 늘려야 할지는 미지수이다. 그렇기에 본 연구에서는 0.3 ~ 0.1 사이의 값으로 학습을 시켰다. 그리고 탄력적인 오류역전과 알고리즘은 계산과정을 보면 한 가지 용접 재료에 있어서 오차를 모두 합산하여 계산하기 때문에 학습률이 그렇게 높지 않고 오차변화율을 통해 탄력적으로 조절되기 때문에 0.1 ~ 0.01사이의 값으로 학습률을 선정하였다.

모멘텀은 앞에서 설명한 것과 같이 일반화된 델타 규칙은 탄력적인 오류역전과 알고리즘처럼 오차 변화율을 사용하지 않기 때문에 사용하고 사용자에 따라 0 ~ 1사이의 값으로 유동적으로 사용된다.

위의 과정을 통해 기본적인 설정이 끝나게 되면 학습을 시작할 수 있다. 이때 몇 가지 주의 사항이 있다. 설계자는 본인이 입력한 학습 데이터와 같은 학습 데이터로 학습시킨 데이터가 있는지 확인하고 만약 기존의 데이터를 사용하지 않고 새로 학습을 시키고 싶다면 데이터를 따로 부를 필요 없이 기존에 저장된 데이터를 불러와 학습을 시킬 수 있다.

위의 과정을 모두 거치고 나면 아래의 그림과 같은 창을 통해 신경망 구성, 학습률 설정, 모멘텀 설정을 한 후 학습을 하게 된다.

항목	CDT-1	CDT-10	CDT-12	CDT-13	CDT-2	CDT-24	CDT-26	CDT-28
1 WM_C	0.2903	0.2903	0.2381	0.3226	0.3548	0.0000	0.2258	0.1613
2 WM_Si	0.5836	0.6143	0.5358	0.5939	0.3788	0.2355	0.0307	0.2833
3 WM_Min	0.2702	0.4136	0.6268	0.7151	0.2408	0.5496	0.2904	0.6066
4 WM_P	0.6250	0.6750	0.5875	0.5125	0.7500	0.5000	1.0000	0.2500
5 WM_S	0.0952	0.0857	0.1619	0.1286	0.0952	0.2857	0.1429	0.3333
6 WM_Ti	0.5357	0.5000	0.6429	0.8929	0.3214	0.3214	0.0000	0.2143
7 WM_B	0.5676	0.7568	0.7568	0.9459	0.4865	0.7027	0.6757	0.7338
8 WM_Al	0.2105	0.3158	0.4211	0.7368	0.0526	0.5789	0.4211	0.9263
9 WM_Cr	0.3582	0.3731	0.3731	0.4030	0.3582	0.3284	0.3881	0.3284
10 WM_Ni	0.0026	0.0026	0.0026	0.0079	0.0026	0.0370	0.0106	0.0212
11 WM_Mo	0.4348	0.4565	0.5652	0.5435	0.4130	0.6087	0.2826	0.6087
12 WM_Cu	0.3214	0.3571	0.3214	0.3571	0.3214	1.0000	0.3571	0.4643
13 WM_V	0.8800	0.9600	0.9600	1.0000	0.8800	0.8800	0.8400	0.8800
14 WM_Nb	0.6667	0.8333	0.8889	0.8889	0.6111	0.8333	0.7222	0.9444
15 WM_Zr	0.1579	0.1579	0.1579	0.2105	0.1579	0.1579	0.0526	0.2632

Fig. 22 Data form for learning

Fig. 22에서 앞서 설정할 입력층, 은닉층, 출력층 및 학습 규칙에 대해 전체적인 신경망을 구성하고 학습률, 모멘텀을 설정할 수 있다. 구성된 신경망의 예시로 Fig. 23과 같은 형태를 가진다.

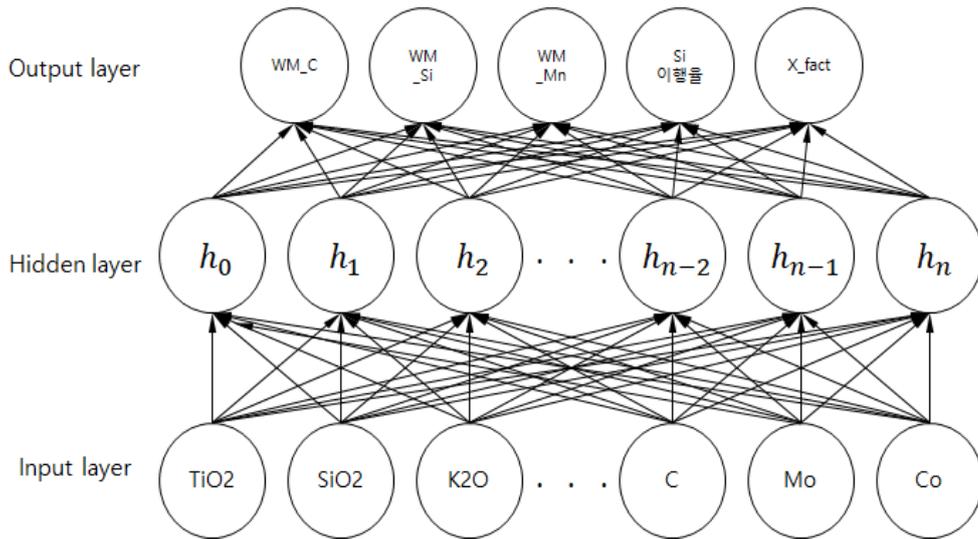


Fig. 23 Network structure for prediction

그 후 학습 버튼을 통해 Fig. 24에서 학습 횟수를 설정한 후 학습을 시작할 수 있다. 학습이 끝나고 나면 Fig. 22에 오른쪽 상단에 저장 버튼을 클릭하면 학습된 모델을 저장할 수 있다.

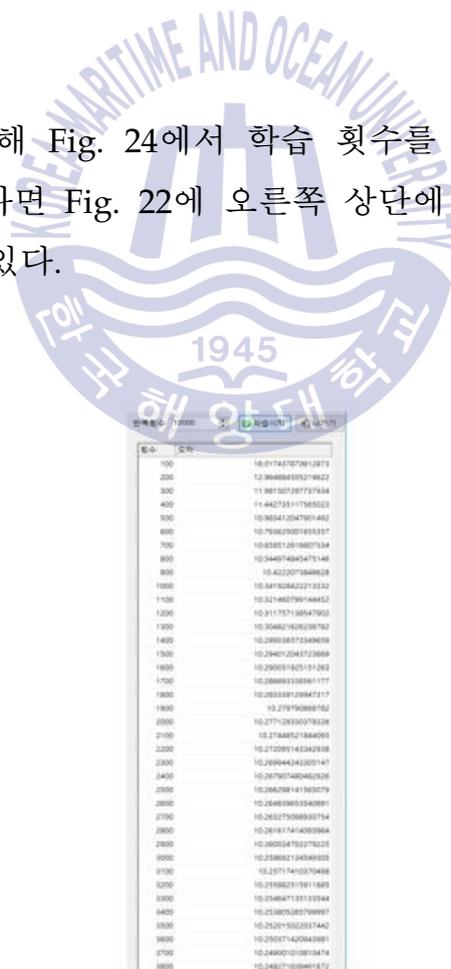


Fig. 24 Form for learning

학습 모델은 총 3개의 시트로 구성되어 있고 하나는 모델의 구성이 저장되어 있는 시트로 모델명, 모델의 목적, 학습률, 모멘텀 등이 저장되어 있고 다른 하나는 출력층의 각 항목에 대한 최소값과 최대값이 저장되어 있고 마지막 시트에서는 학습을 통해 얻어낸 가중치의 값들이 저장되어 있다. 이 학습 모델 데이터를 이용하여 설계에 대한 결과 추정을 하게 되고 그 형식은 Fig. 25과 같다.

A	B	C	D	E	F	G	H	I	J	K	L			
1	모델명	모델목적	중집공정	분류1	입력항목	출력항목	중수	입력중뉴론	은닉1중뉴	은닉2중뉴	출력중뉴론	학습률		
2	22-Rprop-F	22-Rprop-FCAW	산성계	중집결과	중집결과		3	15	15	0	3	0.05		
3														
4	1	A	B	C	D	E	F	G	H	I	J	K	L	
5	2	4916	모델명	변수명	변수유형	minV	maxV	END						
6	3	4920	A	B	C	D	E	F	G	H	I	J	K	L
7	4	4921	1	ID	모델명	CNo	CWValue	END						
8	5	4922	2	22-Rprop-F	22-Rprop-F	C1-0-0	0.970809							
9	6	4923	3	22-Rprop-F	22-Rprop-F	C1-0-1	2.958905							
10	7	4924	4	22-Rprop-F	22-Rprop-F	C1-0-2	-0.70135							
11	8	4925	5	22-Rprop-F	22-Rprop-F	C1-0-3	3.269981							
12	9	4926	6	22-Rprop-F	22-Rprop-F	C1-0-4	4.866514							
13	10	4927	7	22-Rprop-F	22-Rprop-F	C1-0-5	-10.0363							
14	11	4928	8	22-Rprop-F	22-Rprop-F	C1-0-6	-5.93197							
15	12	4929	9	22-Rprop-F	22-Rprop-F	C1-0-7	1.568289							
16	13	4930	10	22-Rprop-F	22-Rprop-F	C1-0-8	-121.4							
17	14	4931	11	22-Rprop-F	22-Rprop-F	C1-0-9	131.8423							
18	15	4932	12	22-Rprop-F	22-Rprop-F	C1-0-10	-20.7623							
19	16	4933	13	22-Rprop-F	22-Rprop-F	C1-0-11	-27.5939							
20	17	4934	14	22-Rprop-F	22-Rprop-F	C1-0-12	-6.34734							
21	18	4935	15	22-Rprop-F	22-Rprop-F	C1-0-13	1.131157							
22	19	4936	16	22-Rprop-F	22-Rprop-F	C1-0-14	2.95963							
23	20		17	22-Rprop-F	22-Rprop-F	C1-0-15	1.61365							
24	21		18	22-Rprop-F	22-Rprop-F	C1-1-0	1.363497							
25	22		19	22-Rprop-F	22-Rprop-F	C1-1-1	0.692265							
26	23		20	22-Rprop-F	22-Rprop-F	C1-1-2	-0.69792							
27	24		21	22-Rprop-F	22-Rprop-F	C1-1-3	-5.52286							
28	25		22	22-Rprop-F	22-Rprop-F	C1-1-4	-3.94887							
	26		23	22-Rprop-F	22-Rprop-F	C1-1-5	-10.6729							
	27		24	22-Rprop-F	22-Rprop-F	C1-1-6	1.474787							
	28		25	22-Rprop-F	22-Rprop-F	C1-1-7	-0.79267							
			26	22-Rprop-F	22-Rprop-F	C1-1-8	-1.94774							
			27	22-Rprop-F	22-Rprop-F	C1-1-9	5.234556							
			28	22-Rprop-F	22-Rprop-F	C1-1-10	1.573381							

Fig. 25 Sheets of learning data

학습 모델까지 저장이 되면 설계 프로그램에 학습 모델을 입력시켜야 한다. 그 이후 신규 설계를 할 때 설계를 끝내고 테스트 전에 결과를 추정해볼 때 원하는 학습 모델을 불러와 추정할 수 있다.

결과를 추정할 때 주의할 점은 학습 모델에 입력된 입력값의 항목만 신규 설계에서 받아 들여 지고 출력값 또한 학습 모델에 입력된 항목만 추정이 가능하다. 그렇기 때문에 설계자는 어떤 학습 모델이 해당 신규 설계에 적절한 추정을 할 수 있을지 잘 파악하고 추정하여야 한다.

앞서 설명한 예측 모델의 기본사항, 가중치 데이터 등 다양한 데이터베이스의 테이블들이 있고 Table 3에 그 관계도가 Fig. 26에 나와 있다.

Table 3 WMANN database table

WMANN 테이블
ANNModel
InpTP
IOParameters
MenuTable
OutTP
WeightValue

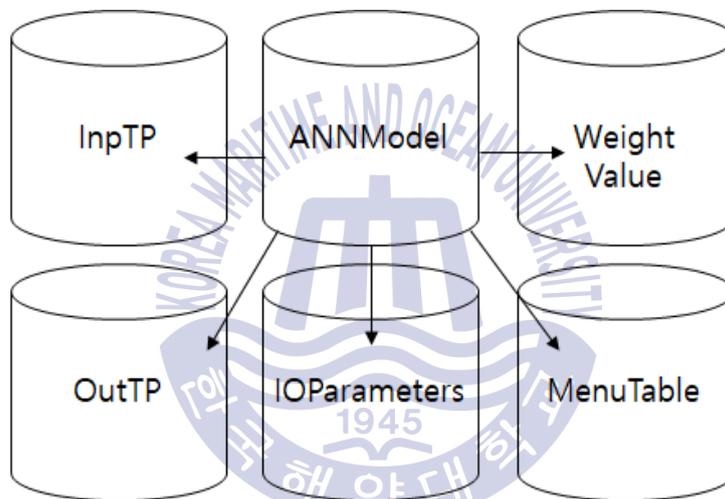


Fig. 26 Relation of WMANN database

ANNModel은 Fig. 21에서 설정한 데이터가 포함되어 있고 예측 모델의 기본적인 사항이 저장된다. IOParameters는 학습 데이터를 생성 시 정규화 과정이 요구된다. 그 과정에서 생성되는 입, 출력 항목에 대한 최소값과 최대값들이 IOParameters 테이블에 저장된다. MenuTable은 학습 데이터의 입력값과 출력값의 분류 체계에 대해 저장이 되어있고 실제 설계과정에서 추정을 할 때 예측 모델에 학습된 분류 체계를 불러오기 위해 사용된다. 마지막으로 InpTp와 OutTP는 예측 모델을 생성하기 위해서는 학습 데이터를 불러와야 하고 이때 입력값과 목표값을 저장하는 테이블이고 예측 모델이 생성되면 InpTp와 OutTP 테이블에 저장된 값은 삭제가 된다.

5.3 인공신경망을 활용한 용접재료 설계지원

학습 데이터를 이용하여 학습을 시킨 후 저장한 데이터를 학습 모델이라고 한다. 학습을 시킬 때는 몇 가지 설정이 필요한데 그 중 학습률이 매우 중요하다. 학습률은 학습 과정 중 오류 역전파 과정에서 가중치 재설정을 위한 오류 값 결정에 반영된다. 학습률은 상황에 따라 값을 설정해야하고 일반적으로 0과 1사이의 값을 사용한다. 학습 데이터가 많지 않다면 굳이 너무 작은 값으로 학습률을 설정하지 않아도 된다. 하지만 데이터 많다면 학습률은 낮게 설정할 필요가 있다. 만약 학습률이 너무 높게 설정되면 데이터가 많기 때문에 가중치가 너무 큰 편차로 변하면 수렴이 안 되거나 너무 일찍 수렴하게 되어 학습이 정상적으로 이루어지지 않을 수 있다. 물론 학습률을 낮게 설정하면 수렴하기까지 더 많은 반복 계산을 요구되어 시간이 오래 걸리는 단점이 있다. 본 연구에서는 학습률을 0.2~0.05 사이의 값으로 설정하였고 데이터의 수가 많을수록 학습률을 낮췄다.

탄력적 오류역전파 알고리즘은 계산 과정에 오차의 변화율을 계산하여 모멘텀으로 사용해서 모멘텀 설정이 필요 없지만 일반화된 델타 규칙은 모멘텀이 필요하기 때문에 설정을 해주어야 한다. 하지만 모멘텀은 학습률과 다르게 특정 수렴구간에 머물러 있지 않게 해주는 역할을 하고 주로 0과 1 사이의 값으로 설정한다.

일반화된 델타 규칙에 대하여 학습률과 모멘텀에 대한 오차값의 변화를 Fig. 27에서 볼 수 있다. Fig. 27에 나온바와 같이 학습률과 모멘텀이 높으면 초기에는 오차값이 급격하게 감소한다. 하지만 이후 학습 횟수가 500,000번이 넘어가는 기점으로 오차값의 감소가 크게 줄어든다. 학습률과 모멘텀을 0.2보다 더 낮은 값으로 설정하면 오차는 줄어들겠지만 추정 정확도는 차이가 거의 없으며 학습에 소요되는 시간이 대폭 들어나는 이유로 학습률과 모멘텀을 0.2로 설정을 하였다. Fig. 28에는 탄력적 오류역전파 알고리즘에 대한 학습률에 따른 오차값의 변화를 알 수 있다.

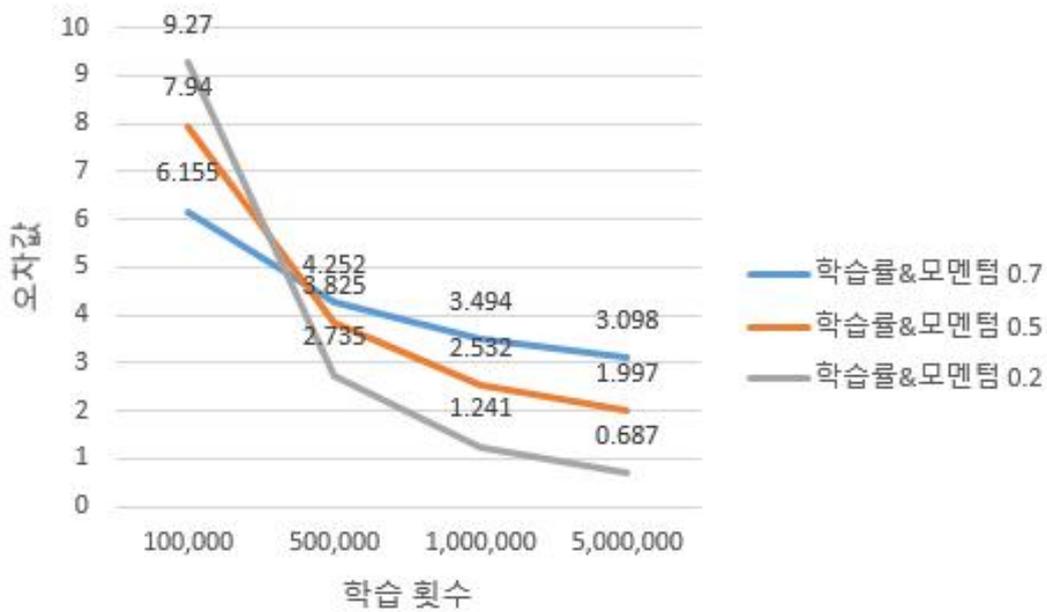


Fig. 27 Variation of error according to learning rate - GDR

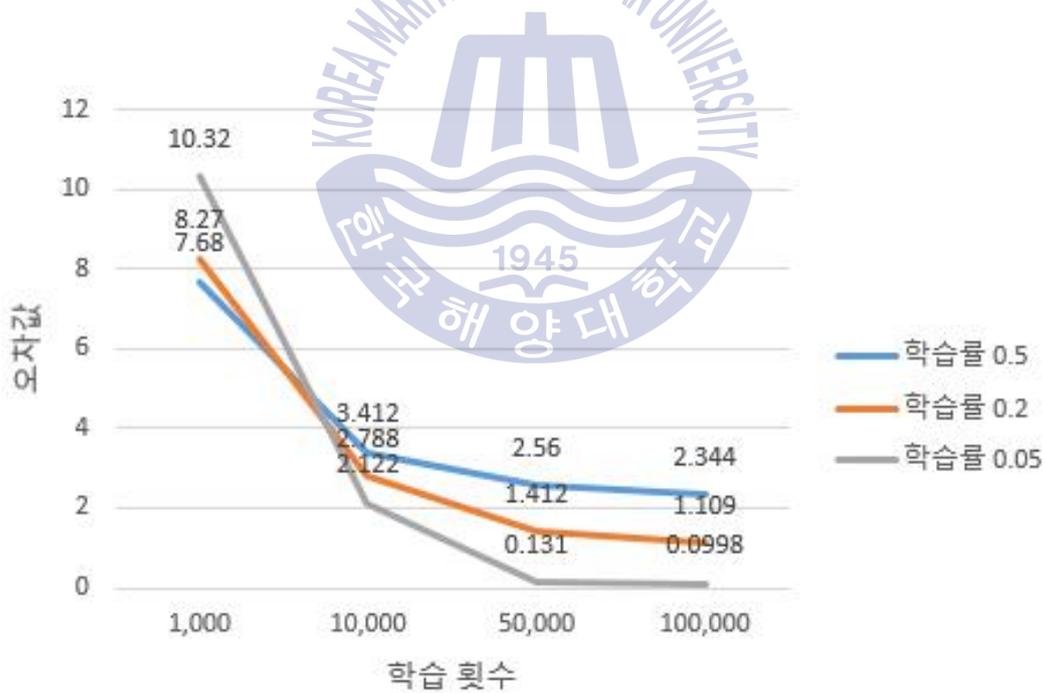


Fig. 28 Variation of error according to learning rate - Rprop

Fig. 28도 Fig. 27과 유사한 경향을 볼 수 있고 일반화된 델타 규칙에 비해 탄력적 오류역전과 알고리즘이 오차값 대비 학습 횟수가 현저히 낮은 걸 확인

할 수 있다. 물론 오차값이 낮다고 무조건 추정 정확도가 올라가는 건 아니지만 일반적으로 오차값이 낮을수록 추정 정확도가 올라간다. 본 연구에 적용한 알고리즘별 학습률과 모멘텀은 Table 4과 같다.

Table 4 Learning rate & momentum used by algorithm

	일반화된 델타 규칙	탄력적 오류역전파 알고리즘
학습률	0.2	0.05
모멘텀	0.2	-

Table 4와 같은 학습률과, 모멘텀을 설정한 이유는 학습률과 모멘텀에 따른 오차와 시간을 고려하여 학습 시 오차값을 최소화시킬 수 있는 값을 찾았다.

은닉층의 뉴런 개수는 일반적으로 입력층의 개수와 같거나 1.2배의 개수로 설정하는데 본 연구에서는 입력층의 개수와 같게 설정하였고 설계 데이터는 여러 공정 중 현재 조선 분야에서 가장 많이 사용하는 FCAW로 선택을 하였다. 학습 데이터에 사용된 입력값과 목표값의 항목의 Table 5와 같다.

Table 5 Classify learning data

입력값	목표값
화학 성분	용착 금속 성분, factor
용착 금속 성분	YP, TS, factor

Table 5의 분류에 따라 학습 데이터를 생성하였다. 또한 Table 5의 분류를 기반으로 제품군 별로 용접 재료 설계 데이터를 모아 학습 데이터를 생성하였다.

학습은 학습 데이터 중 70 ~ 80%의 데이터로 학습을 하고 나머지 데이터를 이용하여 학습 모델의 검증은 시행했다. 학습 모델의 검증은 실제 용접 재료 설계 데이터의 입력값을 학습 모델로 추정하여 실제 목표값과 차이를 비교하였고 본 연구에서 학습한 예측 모델들에 대한 내용은 Table 6, 7과 Fig. 29, 30, 31, 32와 같다.

Table 6 Case of prediction model

	Case 1	Case 2	Case 3	Case 4
학습 데이터 개수	485	91	73	106
입력값의 분류	성분 계산치	성분 계산치	성분 계산치	용착 금속
출력값의 분류	용착 금속, 이행율	용착 금속, 이행율	용착 금속, 이행율	Factor, YP, TS
입력층 노드 개수	32	17	17	15
은닉층 노드 개수	32	17	17	15
출력층 노드 개수	7	4	6	4

Table 7 Detail factor of prediction model

	Case 1	Case 2	Case 3	Case 4
용접 재료의 분류	스테인리스 강-전자세용, 탄소강-저온 강용,전자세용 ,아래보기전용	탄소강-아래 보기전용	탄소강-전자 세용	스테인리스 강-전자세용, 탄소강-저온 강용,전자세용 ,아래보기전용
입력값의 항목	Tio2,Sio2,MgO , ... ,Ni,Mo,Cr	Tio2,Sio2,CaO , ... ,Si,Mn,Mg	Tio2,Sio2,CaO , ... ,Si,Mn,Mg	WM_C,WM_Si , ... ,WM_Ni, WM_Cr

	WM_C, WM_Si		WM_C,	
	, WM_Mn,	WM_Si,	WM_Si,	X_factor,
출력값의	Si이행율,	WM_Mn,	WM_Mn,	J_factor, YP,
항목	Mn이행율,	Si이행율,	Si이행율,	TS
	X_factor,	Mn이행율	Mn이행율	
	J_factor			

Table 6, 7의 내용을 보면 Case 1과 다른 Case의 학습 데이터 수와 입력층, 은닉층의 수의 차이가 많이 나는 것을 볼 수 있다. 이는 특정 분류가 아닌 최대한의 데이터를 사용하여 학습을 시키기 위해 차이가 나는 것이다. Case 2와 3의 입력층의 항목 중 Case1과 4와 다르게 Ni, Cr 등의 성분이 포함되어 있지 않다. 이는 탄소강 분류의 용접 재료에는 해당 화학성분이 사용되지 않거나 매우 소량으로 사용되어 입력층에 포함을 시키지 않았다. Case 4는 입력값을 용착 금속의 성분, 출력값을 기계적 성질로 학습을 하였다. 여기서 용착 금속의 성분은 화학 성분에 'WM_'를 입력한 항목들이고 모재에 와이어가 용융되어 용착된 금속의 성분들을 말한다. 모든 Case에서 출력값의 항목이 적은 경향이 있다. 이는 출력값의 항목인 용착 금속 성분이나 기계적 성질 중 검사결과가 누락되거나 시험을 하지 않은 부분이 많아 데이터의 수집에 한계가 있어 출력값의 항목 수가 많지 않아 데이터 수집의 한계가 있었고 용접 결과의 항목들이 고르지 못해 신뢰도가 높지 않았다.

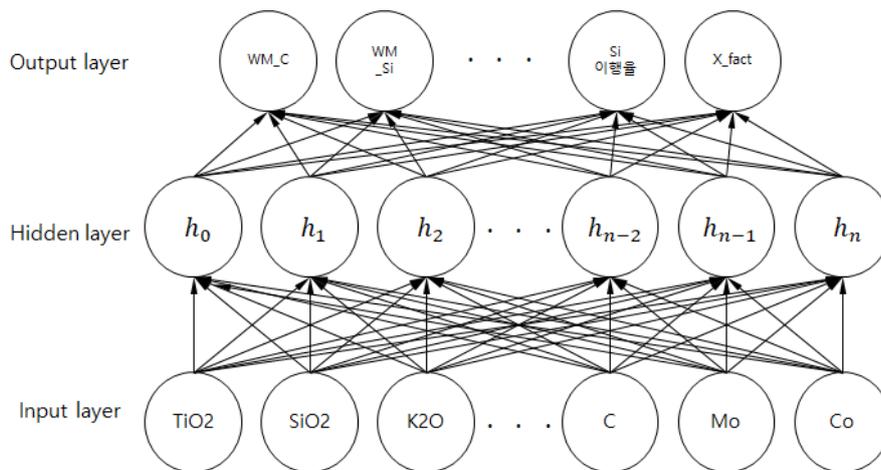


Fig. 29 Network structure for prediction - Case 1

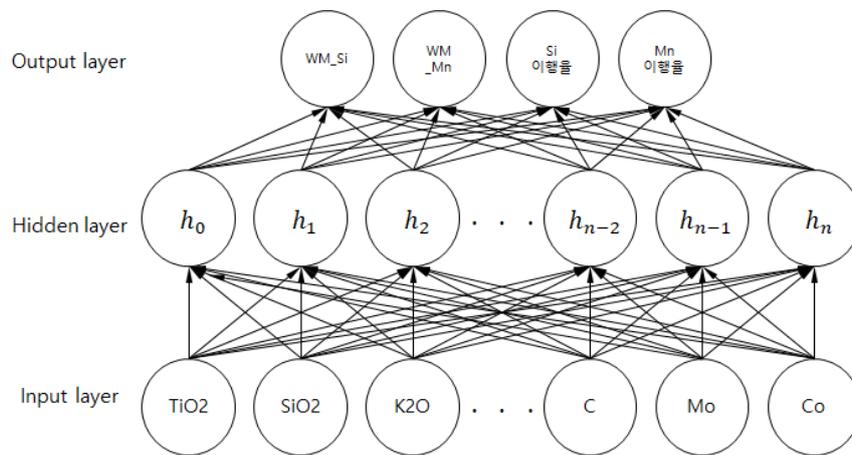


Fig. 30 Network structure for prediction - Case 2

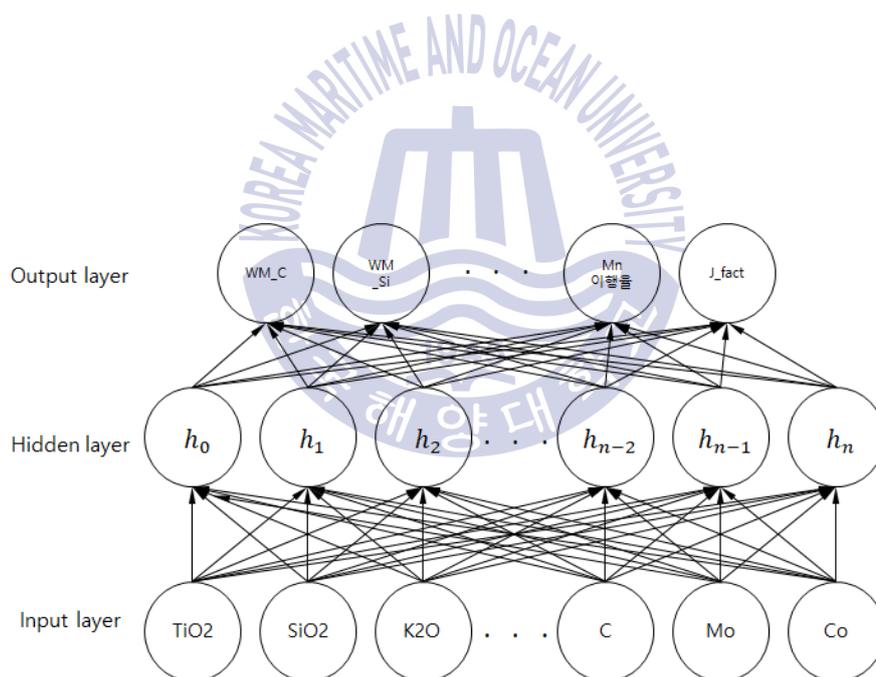


Fig. 31 Network structure for prediction - Case 3

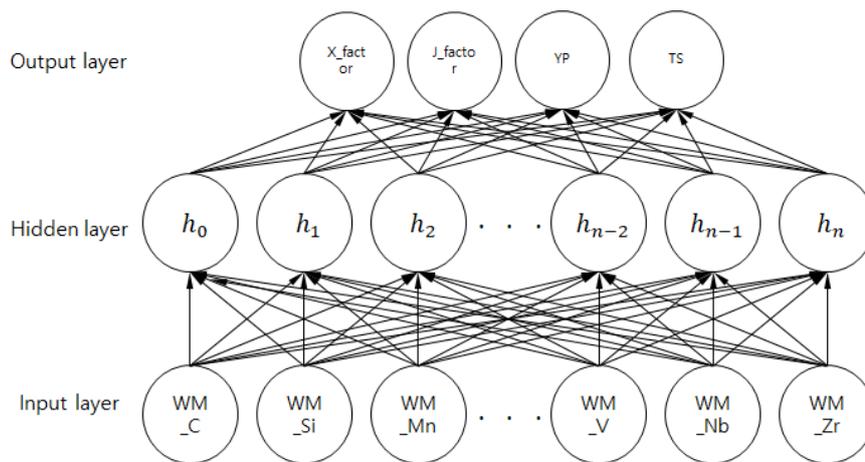


Fig. 32 Network structure for prediction - Case 4

본 연구에서는 모든 Case에 대해 일반화된 델타 규칙과 탄력적 오류역전파 알고리즘 2가지 알고리즘을 사용하여 학습을 하였다.

일반화된 델타 규칙과 탄력적 오류역전파 알고리즘을 사용하여 학습한 데이터에 대한 가중치 값들이 아래 Table 7, 8, 9, 10에 나와 있다.

Table 8 Connect No. & weight value - Case 1

GDR		Rprop	
connect No.	weight value	connect No.	weight value
C1-0-0	1.805933	C1-0-0	2.845786
C1-0-1	1.937155	C1-0-1	-0.57854
C1-0-2	-10.6941	C1-0-2	0.276083
C1-0-3	-6.81494	C1-0-3	-6.21161
C1-0-4	-13.017	C1-0-4	4.836595
...		...	
C2-7-11	4.671521	C2-7-11	-2.09982
C2-7-12	4.396445	C2-7-12	5.416377
C2-7-13	-4.59633	C2-7-13	-0.12289
C2-7-14	-6.48128	C2-7-14	-1.25752
C2-7-15	0.439015	C2-7-15	-0.09804

Table 9 Connect No. & weight value - Case 2

GDR		Rprop	
connect No.	weight value	connect No.	weight value
C1-0-0	-4.81307	C1-0-0	1.265833
C1-0-1	-6.03692	C1-0-1	0.489366
C1-0-2	3.627804	C1-0-2	0.682464
C1-0-3	0.160107	C1-0-3	1.227473
C1-0-4	-35.0323	C1-0-4	8.685423
...		...	
C2-14-27	-4.01378	C2-14-27	0.104486
C2-14-28	0.544359	C2-14-28	-0.57449
C2-14-29	-0.84968	C2-14-29	-1.73318
C2-14-30	-3.66772	C2-14-30	-1.10519
C2-14-31	-0.39114	C2-14-31	0.312384

Table 10 Connect No. & weight value - Case 3

GDR		Rprop	
connect No.	weight value	connect No.	weight value
C1-0-0	-2.25266	C1-0-0	-2.35278
C1-0-1	-3.40398	C1-0-1	0.121229
C1-0-2	-13.791	C1-0-2	3.419709
C1-0-3	3.50694	C1-0-3	-1.87694
C1-0-4	-4.74821	C1-0-4	0.581831
...		...	
C2-5-28	-2.36562	C2-5-28	-0.93773
C2-5-29	-3.41292	C2-5-29	-3.40071
C2-5-30	0.115181	C2-5-30	-0.5645
C2-5-31	1.006435	C2-5-31	1.193131
C2-5-32	0.313369	C2-5-32	-6.03821

Table 11 Connect No. & weight value - Case 4

GDR		Rprop	
connect No.	weight value	connect No.	weight value
C1-0-0	0.416827	C1-0-0	-2.94246
C1-0-1	-4.08262	C1-0-1	-0.91146
C1-0-2	1.237904	C1-0-2	0.242836

C1-0-3	10.18468	C1-0-3	-6.88159
C1-0-4	0.464412	C1-0-4	-1.37331
...		...	
C2-4-11	4.630741	C2-4-11	-0.63286
C2-4-12	1.373362	C2-4-12	-1.47576
C2-4-13	-5.38857	C2-4-13	0.404183
C2-4-14	-5.17382	C2-4-14	2.05838
C2-4-15	0.217712	C2-4-15	-0.91248

Table 8, 9, 10, 11의 가중치 값들은 학습을 통해 계산된 값이고 이를 이용하여 예측 모델의 검증은 시행하였다. 예측 모델 검증은 앞서 서술한 내용에 의해 학습을 시키지 않은 용접 재료 설계 데이터를 예측 모델로 추정된 출력값과 목표값의 차이를 오차로 표현하였다.

Table 8, 9, 10, 11에서 말한 4가지 Case에 대해 학습데이터 개수의 20%의 데이터로 검증을 하였고 그 결과는 Table 12과 같다. Table 12의 평균 오차는 식 (39)를 이용하여 구하였다.

$$MAE(\text{평균절대오차}) = |y_i - \hat{y}_i| \quad (39)$$

식 (39)의 y_i 는 목표값, \hat{y}_i 는 예측값이다. Table 12의 결과를 보면 전체적으로 일반화된 델타 규칙이 탄력적 오류역전과 알고리즘보다 오차가 크게 나오는 경향을 나타냄으로써 탄력적 오류역전과 알고리즘이 일반화된 델타 규칙에 비해 학습 성능이 더 우수하다고 판단했다.

Table 12 Predictive model verification

	Case 1		Case 2		Case 3		Case 4	
	GDR	Rprop	GDR	Rprop	GDR	Rprop	GDR	Rprop
평균	90%							
오차	이상	19.3%	16.6%	8.8%	15.8%	11.7%	18.1%	14.8%

Case 1은 전체 Case 중 학습 데이터가 가장 많은 Case로 일반화된 델타 규칙으로는 거의 학습이 되지 않았는데 이는 데이터 분류상 이질적인 데이터들을

거르지 않고 최대한 모아서 입력 데이터 생성했기 때문이라고 판단된다. 탄력적 오류역전과 알고리즘은 그에 비해 19.3%의 평균 오차를 보였지만 학습이 정상적으로 이루어졌다고 판단하기는 힘들다.

Case 2,3은 학습 데이터가 FCAW의 탄소강의 특정 제품들로 생성하여 Case 2,3에 비해 용접 결과 항목도 비슷한 항목들의 값이 존재했고 이러한 이유로 낮은 평균 오차를 나타내었다고 판단했다.

Case 4는 용접 결과의 항목들 중 용착 금속의 성분을 입력값, 기계적 성질을 출력값으로 학습을 시켰지만 Case 1과 마찬가지로 분류상 이질적인 데이터들을 용접 결과의 항목으로만 학습시키기 위해 생성하여 학습이 정상적으로 이루어지지 않은 것으로 판단된다.



제 6 장 Conclusions

본 연구에서는 FCAW 재료 설계를 지원하기 위해 탄력적 오류역전과 인공신경망을 이용하여 설계 데이터를 학습시켰고 새로운 설계에 대해 결과를 추정해 그 적용성을 검토하였다. 학습에 사용한 알고리즘은 일반화된 델타 규칙과 탄력적 오류역전과 알고리즘을 사용하여 직접 프로그래밍을 했다. 결과 추정에 사용된 데이터는 결과값이 있는 데이터를 이용하여 추정값과 비교를 통해 전체 항목에 대한 평균 오차를 구하여 그 타당성을 검토하였다. 본 연구로부터 다음과 같은 결론을 얻을 수 있다.

(1) 일반화된 델타 규칙은 탄력적 오류역전과 알고리즘보다 예측에 대한 정확도가 떨어지는 것을 확인했다.

(2) 학습 데이터에 사용되는 데이터들의 유사성이 다를수록 학습이 어려워짐을 확인했다.

(3) 학습 데이터에 사용되는 데이터들의 신뢰성, 즉 설계 목표에 맞는 결과 데이터를 이용하면 학습이 잘되는 것을 확인했다.

Reference

Chang-Hyun Kim, Hong-Yeon Yu and Sung-Hoon Hong, 2007, "Adaption of Neural Network Algorithm for Pattern Recognition of Weld Flaws," JOURNAL OF THE KOREA CONTENTS ASSOCIATION, Vol. 7, No. 1, pp. 65~72.

Jin Soo Jung, Hee Keun Lee and Young Whan Park, 2016, "Prediction of Tensile Strength for Plasma-MIG Hybrid Welding Using Statistical Regression Model and Neural Network Algorithm," Journal of Welding and Joining, Vol. 34, No. 2, pp. 67~72.

T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, Accelerating the convergence of the backpropagation method, Biol. Cybernet. 59 (1988) 257–263.

J P. Stagge, B. SendhoJ, An extended Elman net for modeling time series, in: W. Gerstner (Ed.), International Conference on Arti-cial Neural Networks, Lecture Notes in Computer Science, Springer, Berlin, 1997.

T. Tollenaere. Supersab: Fast adaptive backpropagation with good scaling properties. Neural Networks, 3(5), 1990.

W.H. Wolberg, O.L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc. Natl. Acad. Sci. USA 87 (1990) 9188–9192.

M. Toussaint. On model selection and the disability of neural networks to decompose tasks. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2002), pages 245–250, 2002.

M. Riedmiller. Advanced supervised learning in multi-layer perceptrons - From backpropagation to adaptive learning algorithms. Computer Standards and Interfaces, 16(5):265-278, 1994.

