

다양한 제조시스템에서 생산률 추정을 위한 지도학습 기반의 DNN 모델 활용

문덕희[†] · 쩌안안 · 김동욱 · 신양우

Application of DNN Models with Supervised Learning for Estimating Production Rates in Various Manufacturing Systems

Dug Hee Moon[†] · Quynh Anh Trinh · Dong Ok Kim · Yang Woo Shin

ABSTRACT

In order to optimize the manufacturing system design problems, a model that can estimate the performance of the system is required. Various methodologies such as mathematical models, simulations, and meta models have been used for the estimation. Recently, artificial intelligence models have become a new alternative. In this paper, deep neural networks (DNN) models are applied for analyzing the accuracies of the estimating production rates in serial lines and assembly lines. Simulation models are developed for the systems, and the results obtained through experiments are used for learning processes for finding optimal DNN models. In the optimizing processes, combinations of various parameters such as activation function, the number of layers, and the number of nodes were considered. In addition, a method to increase the accuracy of the DNN model was presented by adding extreme data set to the learning process, which is known to be less accurate when using other methodologies.

Key words : Manufacturing systems, Production rates, Deep neural networks, Supervised learning, Activation functions

요약

제조시스템 설계문제에서 최적화를 수행하기 위해서는 시스템의 성능을 추정할 수 있는 모델이 필요하다. 이를 위해 수학적 모델, 시물레이션, 메타모델 등 다양한 방법론이 활용되고 있는데, 최근에는 인공지능 모델의 활용이 확대되고 있다. 본 논문에서는 지도학습을 이용한 심층신경망(DNN) 모델을 이용하여 직렬라인, 조립라인 등의 생산률 추정 정확성에 대해 분석을 하고자 한다. 각 시스템에 대해 시물레이션 모델을 개발하였고, 실험을 통해 얻은 결과를 이용하여 학습을 수행함으로써 최적의 DNN 모델을 개발하였다. 이 과정에서 활성화 함수, 은닉층의 개수, 노드의 개수 등 다양한 모수들의 조합을 고려하였다. 또한 다른 방법론을 사용하였을 때 정확도가 떨어진다고 알려진 극단적 데이터를 학습과정에 추가함으로써 DNN 모델의 정확성을 높이는 방법을 제시하였다.

주요어 : 제조시스템, 생산률, 심층신경망, 지도학습, 활성화함수

1. 서론

1.1 연구배경 및 목적

제조시스템 설계과정에서 작업장의 구조는 제조환경

* 이 논문은 2023~2024년도 창원대학교 자율연구과제 연구비 지원으로 수행된 연구결과임.

Received: 11 October 2024, **Revised:** 13 December 2024, **Accepted:** 17 December 2024

† Corresponding Author: Dug Hee Moon

E-mail: dhmoon@changwon.ac.kr

Changwon National University Department of Industrial and Systems Engineering

의 특성을 반영하여 흐름라인(Flow Line 또는 Serial Line), 병렬라인(Parallel Line), 병렬흐름라인(Flexible Flow Line), 개별작업장(Job Shop)은 물론 소수의 작업자가 전체공정을 담당하는 셀(Cell) 라인이 될 수도 있다(Gershwin, 1994; Lee et al., 2020b). 한편, 서로 다른 부품이 외부에서 조달되거나, 내부의 다른 라인에서 만들어진 후 조립이 되는 조립라인(Assembly Line) 형태도 가능하다(Gershwin, 1994; Lee et al., 2020b). 더 나아가 전체를 하나의 라인으로 연결하여 구성할 수도 있고, 여러 개의 부분라인(Sub-Line)으로 분리하여 구성할 수도 있다(Lee et al., 2020b).

이와 같이 제조시스템 설계 문제의 최적해를 찾기 위해서는 두 단계가 필요하다. 첫 번째 단계는 시스템의 성능(Performance)을 예측하는 모델을 개발하는 단계이고, 두 번째 단계는 개발된 예측모델을 이용하여 최적화를 수행하는 단계이다(Koyuncuoğlu, 2024; Lee et al., 2020a).

성능예측모델은 대기행렬망이론(Queueing Network Theory), 확률과정모델(Stochastic Process Model)과 같은 수학적 모델(Mathematical Model)을 이용하는 방법, 시뮬레이션(Simulation) 모델을 이용하는 방법 등이 기본적으로 사용되고 있다(Gershwin, 1994). 또한 시뮬레이션 실험을 통해 얻은 데이터를 이용하여 메타모델(Meta-model)을 개발하는 방법도 사용되고 있다. 메타모델은 회귀모형(Regression Model)과 같은 수학적 모델을 사용하는 방법이 있고, 최근에는 인공지능(Artificial Intelligence) 모델을 사용하는 연구들이 늘고 있다.

이와 같은 방법론 중에 가장 이상적인 방법은 수학적 모델을 이용하는 것이다. 수학적 모델을 이용하면 시스템의 성능을 정확히 평가할 수 있으며, 계산 속도도 빠르다는 이점이 있다. 하지만 복잡한 제조시스템에 대해서는 모델을 만드는 것도 제한적이고, 실사 모델을 만든다고 하더라도 정확한 해를 구하는 과정이 어렵기 때문에 근사해법을 사용해야 한다(Gershwin, 1994; Shin과 Moon, 2017; Papadopoulos et al., 2019). 하지만 근사해법을 통해 얻은 값은 참값이 아니다.

반면에 시뮬레이션은 다양한 제약이 존재하는 복잡한 시스템에 대해 모델을 개발하고 분석하는 것이 가능하므로 참값에 대한 추정치를 실험을 통해 쉽게 얻을 수 있다. 하지만 최적화 과정에 시뮬레이션 모델을 직접 사용하는 경우에는 의사결정변수의 값이나 모수의 값이 변경될 때마다 매번 모델을 변경하고, 실험을 다시 수행해야 하는 등 수학적 모델보다 많은 시간이 필요하다(Gershwin, 1994; Koyuncuoğlu, 2024; Lee et al., 2020a). 더 나아가 최적해 근처에서는 시뮬레이션 결과값의 임의효과(Random Effect)로 인해 수렴속도가 늦어지는 단점도 존재한다.

예를 들어 10대의 기계와 9개의 버퍼(Buffer)로 구성된 직렬라인에 대해 버퍼최적할당문제(BAP: Buffer Allocation Problem)을 해결하는 상황을 가정해 보자. 각 버퍼의 크기가 0개부터 10개까지 가능하며, 총 버퍼의 수량이 50개인 경우 최적화방법으로 버퍼 1개씩 차례로 할당하는 탐색(Search)방법을 선택한다면 시뮬레이션 실험을 해야 할 최악의 횟수는 $1+9^{50}$ 이 될 수도 있다. 물론 어떤 최적화 방법을 선택하느냐에 따라 경우의 수는 줄어들겠지만 많은 횟수의 실험을 해야 하는 것은 명확하다.

위와 같은 두 가지 방법론의 한계점을 극복하기 위해 제안된 방법이 메타모델(Meta-model)이다. Can과 Heavey (2011)는 특히 이산사건 시뮬레이션(Discrete Event Simulation)을 활용하는 메타모델 방법론이 최적화 문제가 요구하는 정확성과 소요시간을 동시에 만족하는 좋은 대안이 될 수 있다고 하였으며, Khayyati와 Tan(2022b)은 이 방법이 좀 더 넓은 범위로 시스템을 확장시키는 경우에도 효율적이라고 하였다.

본 연구에서는 Figure 1과 같이 제조 시스템 설계 문제에서 생산률(Production Rate 또는 Throughput)을 추정할 목적으로 시뮬레이션을 이용하여 얻은 데이터를 활용하여 심층신경망(DNN) 모델을 개발하는 방법의 타당성에 대해 논의한다. 분석 대상은 제조시스템설계분야의 대표적 대상시스템인 고장이 발생하는 직렬라인, 고장이 발생하는 직렬라인을 부분라인(Sub-line)으로 하는 조립라인, 부분라인 내에는 공정간 버퍼가 존재하지 않는 특수한 경우의 조립라인 등으로 한다.

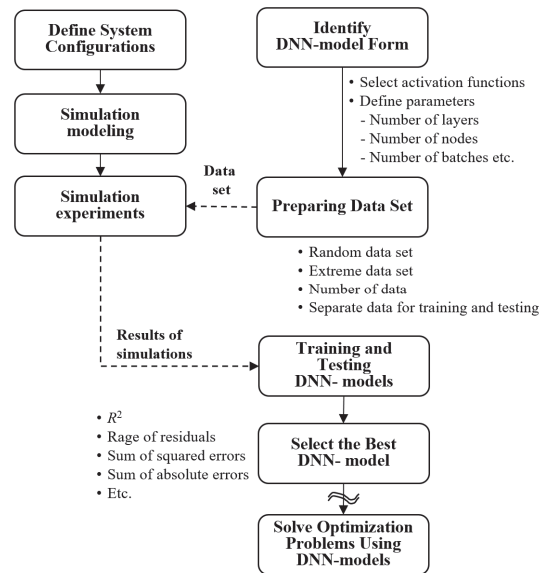


Fig. 1. Approach for combining simulation and DNN model

DNN 모델을 개발하기 위해서 널리 알려져 있는 활성화 함수를 비교함은 물론, 은닉층(Hidden Layer)의 개수, 노드(Node)의 개수, 학습에 사용되는 데이터 배치(Batch)의 크기 등이 인공지능망 모델의 적합성에 미치는 영향 등도 실험을 통해 분석하고자 한다. 마지막으로 DNN 모델을 이용한 메타모델 방법론과 표면반응법(RSM)을 이용한 회귀모형 기반의 메타모델 방법론의 정확성을 비교함

으로써 학습을 위한 데이터의 크기 등에 대해 논의한다.

1.2 관련 연구

제조시스템 설계문제를 수학적모델링 방식으로 접근한 연구들은 지난 수십년간 수행되었다. 초기 연구들은 Gershwin(1994), Li와 Meerkov(2009)의 저서에 상세히 소개되어 있으며, Papadopoulos et al.(2019)의 논문에도 최적버퍼할당문제를 해결하기 위한 방법론 별로 잘 정리되어 있다. 그리고 시뮬레이션 기법을 이용해 제조시스템을 분석하는 주제는 워낙 광범위하므로 본 논문에서는 별도로 언급하지 않는다.

시뮬레이션과 메타모델링 방법을 결합하여 제조시스템 설계문제를 해결하고자 한 연구들이 비교적 근래에 수행되고 있는데, 회귀모형 기반의 메타모델을 이용해 접근한 선행연구들은 Can과 Heavey(2011), Lee et al.(2020a, 2020b), Moon et al.(2021) 등에 잘 소개되어 있다(표현의 편의상 앞으로 메타모델은 회귀모형 기반의 메타모델을 지칭하는 것으로 한다). 이 연구들은 먼저 시스템을 시뮬레이션 모델로 만들고, 실험계획법(DOE, Design of Experiments)을 이용하여 최소의 시뮬레이션 실험을 설계하고, 그 결과를 회귀모형 개발에 활용함으로써 시스템의 성능을 예측하는 방법을 사용하였다. 이 방법의 장점은 모델 개발에 필요한 시뮬레이션 실험 횟수를 줄일 수 있다는 점이다.

Lee et al.(2020b)은 혼류생산과 Modular 공법을 이용한 자동차 차체공장의 부분모델에 대해 메타모델을 이용해 생산률을 최대화시키는 버퍼할당문제를 다루었으며, Lee et al.(2020a)은 Layered 공법을 적용하는 자동차 차체공장의 부분모델에서 목표생산률을 만족시키면서 버퍼 투자비용, 가용률을 고려한 장비 도입비용, 재공품 재고 유지비용, 그리고 부품이송설비 투자비용의 합을 최소화시키는 비용 최소화 문제를 최적화 하였다. Moon et al.(2021)은 내연기관 자동차와 전기자동차를 동시에 생산하고, 두 종류 자동차의 하부구조물(Under Body) 생산방식이 흐름라인과 셀(Cell)방식으로 서로 다르며, 전기자동차의 생산비율이 점차 증가할 때 하부라인 구조의 재구성이 가능한 차체공장 전체모델에 대해 생산률의 최대화, 이익의 최대화 등을 다루는 문제를 시뮬레이션을 활용한 메타모델링 방식으로 접근하였다.

한편, AI 기술의 확산과 더불어 기계학습(Machine Learning)을 제조시스템 전반에 활용하는 추세가 증가하였는데, Fonseca와 Navarrese(2002)는 신경망모델을 시뮬레이션에 연동하여 Job shop 환경에서 MLT(Mnufacturing

Lead Time)을 예측하는데 활용하였다. Kang et al.(2020)은 생산라인의 설계 및 운영 주제에 대해 기계학습을 이용한 과거 연구들을 분류한 논문을 발표하였다. Khayyati와 Tan(2022b)은 지도학습을 이용하여 단일공정 문제에서 도착시간간격과 공정시간의 상호관계를 규명하는 연구를 하였으며, Khayyati와 Tan(2022a)은 작업장에서 실시간으로 얻는 데이터를 기계학습 방법으로 분석하여 적절한 기계 할당 등 운영정책을 변경하는 방법에 대한 연구를 발표하였다. 특히 Can과 Heavey(2012), Khayyati와 Tan(2022b)은 ANN(Artificial Neural Networks) 또는 DNN(Deep Neural Networks)과 같은 방법론을 이용하여 제조시스템 설계문제를 분석할 때 여러 학습방법 중에 지도학습(Supervised Learning) 방법이 적합하다고 하였다.

본 연구에서는 제조 시스템 설계 문제에서 생산률을 추정할 목적으로 지도학습을 이용한 DNN 모델을 적용할 경우의 효과에 대해 논의해 보려고 한다. 이 논문은 Moon et al.(2022)에서 발표된 내용을 보완한 것이다. 주요 보완 내용은 시스템의 종류를 고장이 발생하는 직렬라인, 고장이 발생하는 직렬라인을 부분라인(Sub-line)으로 하는 조립라인, 서브라인 내에는 공정간 버퍼가 존재하지 않는 특수한 경우의 조립라인 등 3개로 확대하였다. 또한 다양한 활성화 함수를 추가하였으며, 은닉층의 개수, 노드의 개수, 학습에 사용되는 데이터 배치의 크기도 더 다양한 값으로 설정하였다. 마지막으로 DNN 모델을 이용한 메타모델 방법론과 표면반응법(RSM)을 이용한 회귀모형 기반의 메타모델 방법론의 정확성을 비교함으로써 학습을 위한 데이터의 구성방법, 데이터의 크기 등에 대해 논의한다.

2. 분석을 위한 제조시스템 유형 정의

본 연구에서 검토해 볼 제조시스템은 다음과 같다. 먼저 ①제조시스템 설계분야에서 가장 기본적으로 사용되는 고장이 발생하는 직렬라인, ②부분라인이 직렬라인이면서 조립이 수행되는 조립라인, ③부분라인 내에는 버퍼가 허용되지 않고 혼류생산이 허용되는 좀 더 복잡한 조립라인 등 3종류의 시스템을 선택하여 차례로 분석하였다. 특히 세 번째 시스템의 경우 수학적 모델로 접근하는 것은 매우 어렵다.

이들 시스템의 상세 구조 및 특성은 다음과 같으며, 이와 같은 가정을 토대로 각각 시뮬레이션 모델을 개발하여 생산률을 계산하였다. 이때 생산률은 단위시간당 생산

된 제픁의 수량으로 정의하는데(Gershwin, 1994), 시뮬레이션을 통해 얻은 생산량을 시뮬레이션시간 (통계량 수집시간)으로 나눈 값으로 계산한다, 그 이유는 모든 공정의 공정시간 평균을 1(단위시간)으로 정의하였기 때문이다.

2.1 직렬라인

본 연구에서 고려하는 직렬라인은 아래 Figure 2와 같으며, 다음과 같은 특성을 가진다.

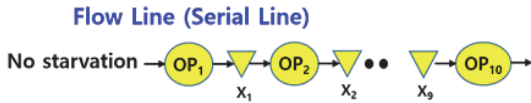


Fig. 2. Serial line with 10 machines (11 variables)

- 직렬라인은 10개의 공정으로 구성되며, 각 공정에는 1대의 기계가 있다.
- 공정과 공정 사이에는 버퍼가 존재하고, 버퍼의 크기($x_1 \sim x_9$)는 0개~8개까지로 유한하다. 다만 공정 1에는 항상 충분한 부품이 존재해서 기근현상이 발생하지 않는다(No-starvation).
- 각 기계는 독립적으로 고장이 발생한다(Unreliable Machines). 고장시간간격 및 수리시간에 관한 분포는 지수분포(Exponential Distribution)를 따른다. 이때 모든 기계의 독립효율($x_{10} = \rho$, Isolated Efficiency)은 0.94~0.98의 범위를 가지며, MTTF (Mean Time to Failure, x_{11})는 160~320분 사이의 값을 가지고, MTTR(Mean Time to Repair)은 ρ 와 MTTF가 결정되면 아래 식(1)에 의해 자동으로 결정된다.

$$\rho = \frac{MTTF}{MTTF + MTTR} \quad (1)$$

- 막힘현상은 BAS (Blocking After Service)를 가정하며, 고장은 Operation Dependent Failure를 가정한다.
- 각 기계에서의 공정시간은 평균이 1인 지수분포를 따른다.

2.2 조립라인

본 연구에서 고려하는 조립라인은 아래 Figure 3과 같으며, 다음과 같은 특성을 가진다.

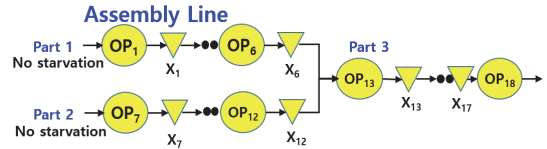


Fig. 3. Assembly line with 18 machines (19 variables)

- 전체라인은 3개의 부분라인(Sub-lines)으로 구성되어 있으며, 각 라인에는 6개의 공정이 직렬로 연결되어 있고, 각 공정에는 1대의 기계가 있다. 부품 1과 부품 2는 해당 부분라인에서 생산된 후 공정 13(OP13)에서 조립이 수행된다.
- 공정과 공정 사이에는 버퍼가 존재하고, 버퍼의 크기는 0개~5개까지로 유한하다($x_1 \sim x_5, x_7 \sim x_{11}, x_{13} \sim x_{17}$). 다만 조립을 위한 버퍼(x_6, x_{12})의 크기는 1개~10개로 한다. 또한 공정 1과 공정 7에는 항상 충분한 부품이 존재해서 기근현상이 발생하지 않는다.
- 각 공정의 공정시간과 고장관련 분포(x_{18}, x_{19})는 (2.1절 직렬라인에서 가정한 것과 동일하다.

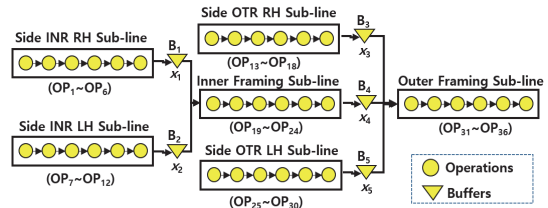


Fig. 4. Special assembly line (Lee et al.(2020a), 9 variables)

2.3 특수한 형태의 조립라인 (차체공장)

본 연구에서 고려하는 특수한 형태의 조립라인은 Lee et al.(2020a)에서 고려하였던 시스템으로 Layered Build 방식을 적용하는 자동차 차체공장의 일부분을 대상으로 하는데, 그 형태는 Figure 4와 같으며, 다음과 같은 특성을 가진다.

- 전체라인은 6개의 부분라인(Sub-lines)으로 구성되어 있으며, 각 부분라인에는 6개의 공정이 직렬로 연결되어 있고, 부분라인 내에 공정간 버퍼는 없다.
- 조립은 두 곳에서 이뤄지며, 조립을 위한 부분라인간의 버퍼(그림에서 $B_1 \sim B_5$)만 존재하는데, 버퍼의 크기($x_1 \sim x_5$)는 1~20개이다.
- 각 공정의 공정시간과 고장관련 분포(x_6, x_7)는 2.1절 직렬라인에서 가정한 것과 동일하다. 다만 TTF

(Time to Failure)는 Time Dependent Failure를 가 정한다.

- 이 라인에서는 두 종류의 제품을 혼류로 생산하 는 데, 혼류비율(x_8)은 0~1사이의 값을 가진다.
- 한 차종(대표 차종)에 대해 작업장 별로 공정 수행 에 소요되는 시간(PT: Processing Time)은 동일한 상수 값을 갖는 것으로 가정한다. 두 번째 차종의 PT를 1로 할 때 대표차종의 PT(x_9)는 0.9~1 사이 의 값을 가지는데, 모든 공정에서 동일하다.
- 부분라인 내의 공정간 이송은 서브라인 내의 모든 공정에서 작업이 완료된 제품이 동시에 다음 공정으 로 이동하는 동기화이송방식(Synchronized Transfer) 을 가정한다.

위와 같은 상황에서 Lee et al.(2020a)은 생산률과 각 버 퍼의 평균수량 등을 추정하기 위한 메타모델을 2차 회귀식 을 이용하여 정의하였으며, DOE(Design of Experiments) 를 활용하여 156개의 데이터를 결정한 후 시뮬레이션 실험 을 통해 얻은 생산률을 이용하여 회귀모델을 결정한 후 에 최적화 과정에 활용하였다.

3. 연구방법론

본 연구에서 사용하는 DNN 모델의 구조는 Figure 5 와 같다. DNN 모델에서는 여러 개의 은닉층을 활용하여 학습을 진행하는데, 파이썬(Python)을 이용하여 모델링 을 하였다. 이때 적합한 모형을 결정하기 위해 다음과 같 은 요인들을 고려하여 최적화해야 한다.

- 노드(Node)의 개수
- 은닉층의 개수
- 학습을 수행하는 데이터 बै치(Batch)의 수
- Epoch 횟수
- 활성화 함수 (Activation Function)

이 이외에도 최적화를 위한 방법(Optimizer), 평가를 위한 척도(Loss Function) 등도 결정해야 한다.

AI 모델에 활용되는 활성화 함수는 매우 다양한 방법 들이 개발되어 있는데(Apicella et al. 2021; Dubey et al., 2022; Nwankpa et al., 2018; Ramachandran et al. 2017), 이들 중 사전 실험을 거쳐 ELU, ReLU, Sigmoid, Softmax, Softplus, SoftSign 등 6종류의 활성화 함수를 최종적으로 사용하였다. 각 함수의 형태 및 수식은 지면 관계상 생략한다.

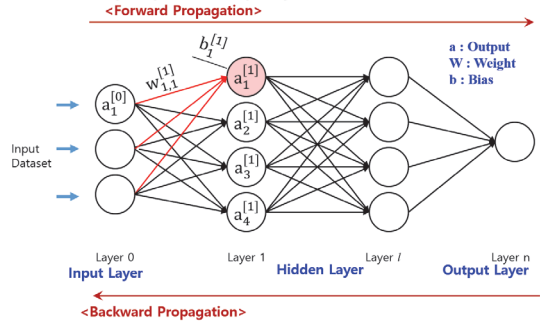


Fig. 5. Structure of DNN Model

DNN 모델에 대해 지도학습을 수행하기 위해서는 선 택된 변수들의 값에 대해 생산률이 얼마인지 알아야 한 다. 따라서 생산률의 참값을 추정하기 위한 방법으로 시 물레이션을 활용하였다. 물론 시뮬레이션을 이용해 얻은 값이 이론적인 참값이라고 할 수는 없지만 참값을 모르 는 상황에서 대체할 수 있는 유용한 방법이다.

본 연구에서 고려하는 시스템에 대해서는 두 종류의 시뮬레이션 도구를 사용하였는데, 직렬라인과 조립라인 의 경우 Siemens사의 Plant Simulation을 사용하였고, 특 수한 형태의 조립라인(차체공장)은 선행연구인 Lee et al. (2020a)와의 일관성 유지를 위하여 Rockwell사의 ARENA 를 이용하여 시뮬레이션 모델을 개발하였다. 시뮬레이션 모델의 개발 환경과 실험조건은 아래 Table 1과 같다.

Table 1. Comparison of simulation conditions

	Serial Lines	Assembly Lines	Body Shop
Modeling Tools	Plant simulation	Plant simulation	ARENA
Warmup period	10,000	10,000	30,000
Data Gathering Time	100,000	100,000	300,000
Replications	10	10	20
Failure	ODF ¹⁾	ODF ¹⁾	TDF ²⁾
Blocking	BAS ³⁾	BAS ³⁾	BAS ³⁾

¹⁾ ODF : Operation Dependent Failure

²⁾ TDF : Time Dependent Failure

³⁾ BAS : Blocking After Service

개발된 DNN 모델의 정확성을 평가하기 위해서는 평 가함수가 필요하다. 먼저 최적의 DNN 모델을 결정하기 위하여 식(2)와 같이 DNN 모델의 설명 능력을 평가하는 R^2 값을 이용하였다.

$$R^2 = \frac{SSE}{SST} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2)$$

$$= 1 - \frac{SSR}{SST} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

부수적으로 각각의 개별 평가데이터에 대해 아래 식 (3)과 같이 상대오차(Relative Error)를 계산하였다. 이때, y_i 는 시뮬레이션 실험을 통해 얻은 생산률값(PR_{sim})이며, \hat{y}_i 는 DNN 모델을 통하여 얻은 생산률값(PR_{DNN})으로 정의한다. 이와 같은 방법으로 얻은 상대오차값을 평가데이터 전체에 대하여 평균(Average), 최대값(Maximum), 최소값(Minimum), 최대값과 최소값의 차이인 범위(Range)를 계산하여 DNN 모델의 정확성을 평가하는데 보조수단으로 활용하였다.

$$R_Error_i (\%) = \frac{(\hat{y}_i - y_i)}{y_i} \times 100 \quad (3)$$

4. 실험결과

4.1 직렬라인

본 연구에서 고려하는 직렬라인은 Figure 2와 같이 10개의 공정으로 공간간 버퍼 9곳에 대한 변수와 MTTF, MTTR 관련 변수 2개 등 총 11개의 변수로 구성되어 있다. 최적의 학습모델을 얻기 위해 Table 2와 같이 모수들의 종류를 정하였다. 노드의 수와 Epoch의 수는 각각 22개와 2000회로 고정하였으며, 최적화방법은 ADAMS를 사용하였다. 나머지 모수들은 범위를 주어 실험을 하였는데, 총 실험 조합의 수는 $10 \times 5 \times 7 = 350$ 개다.

Table 2. Combinations of experiments for serial line

Parameters	Values
Number of Nodes	22
Number of Epochs	2000
Number of Layers	1, 2, 3, .. 9, 10
Number of Batches	8, 16, 32, 64, 128
Activation Functions	Softmax, SoftSign, ELU, ReLU, Softplus, Sigmoid,
Optimizer	ADAM

4.1.1 실험설계

직렬라인의 생산률을 추정하기 위해 DNN 모델이 적합한지 분석하기 위하여 Figure 6과 같은 실험절차를 설계하였다.

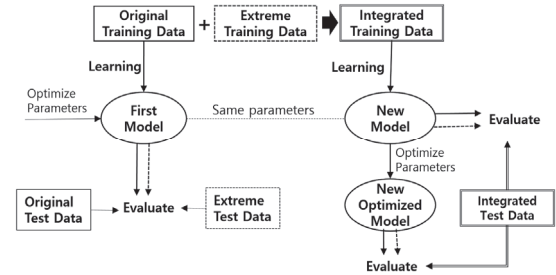


Fig. 6. Structure of DNN Model

최적 학습모형 결정을 위해 사용할 데이터는 두 종류를 준비하였다. 우선 각 변수의 범위내에서 일양(Uniform) 분포를 가정하고 임의로(Random) 추출하여 1000개의 데이터 세트를 만들었다. 이를 최초데이터 세트(Original Data Set)라고 부른다. 그 중에서 800개의 데이터를 임의로 추출하여 학습에 사용하였으며, 나머지 200개의 데이터를 평가에 사용하였다.

제조시스템의 생산률을 추정하는 선행연구들에서 수학적 근사해법을 사용할 경우나 회귀모형을 이용한 메타 모델을 사용할 경우 모두 라인의 길이가 길거나 크기가 0인 버퍼가 많은 경우 정확도가 떨어진다고 하였다(Gershwin, 1994; Lee et al. 2020b; Shin과 Moon, 2017). 따라서 DNN 모델에서도 유사한 상황이 발생하는지, 만일 발생한다면 추가 학습을 통하여 정확도를 높일 가능성이 있는지 확인하고자 추가로 데이터 세트를 마련하였다. 최초 데이터 세트는 변수의 범위내에서 임의로 추출하였지만, 추가 데이터 세트는 변수($x_1 \sim x_9$)의 생성확률을 Table 3과 같이 정하여 각 변수의 양측 극단 값이 많이 포함된 200개 생성하였다. 이를 편이상 극단 데이터 세트(Extreme Data Set)라 부른다. 이중 임의로 100개는 평가용으로 하고, 나머지 100개는 차후에 최초데이터 800개에 추가하여 900개 데이터로 학습을 시켰다. Table 4는 직렬라인의 DNN 모델 개발에 활용한 임의 데이터 세트와 극단 데이터세트의 사례다. 이 데이터들은 학습 및 평가에 활용될 때 0~1사이의 값으로 정규화시켜 DNN 모델의 입력자료로 사용한다.

Table 3. Probability for generating extreme data set

Buffer Size ($x_1 \sim x_9$)	0	1	2	3	4	5	6	7	8
Probability (%)	30	10	5	5	5	5	5	5	30

4.1.2 1차 실험

1차 실험에서는 최초데이터 800개를 이용하여 학습을 수행하고, 최적화 과정을 통해 결정된 DNN 모델을 이용하여 평가용 최초데이터 200개와 평가용 극단데이터 100개에 대한 평가를 각각 실시하였다. 최적화 과정은 6종류

의 활성화함수별로 60개의 모수조합에 대해 DNN 모델을 학습시켜 가장 우수한 조건을 찾아내는 과정이다. 이와 같은 과정을 거쳐 도출된 학습모델의 정확도 분석 결과는 Table 5에 제시되어 있다.

먼저 평가용 최초데이터를 분석한 결과 R^2 측면에서는 활성화 함수를 Softmax로 하였을 때 R^2 값이 99.52%로 가장 우수하였고, 다음으로 Softplus, ELU를 사용할 때 우수한 결과를 보여주었다. 일반적으로 상대오차의 평균 값은 0에 가깝고, 범위가 작을수록 바람직한데 평균은 Softplus가 가장 근접하였다.

Table 4. Sample data set used for developing DNN model of serial line (before normalizing)

	Data		Buffer Size									x_{10} (ρ)	x_{11} (MTTF)	PR (sim)
	No.	Use [†]	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9			
Original (Random) Data Set	237	T	1	6	1	3	0	5	2	2	2	0.942	205	0.496718
	339	T	3	0	3	2	3	0	8	7	7	0.940	226	0.499398
	748	T	0	5	8	3	1	2	6	7	2	0.947	163	0.572045
	892	E	2	8	4	2	6	8	8	0	8	0.965	164	0.530911
Extreme Data Set	1007	T	8	8	0	8	0	2	0	7	0	0.947	204	0.488551
	1030	T	0	0	0	0	1	8	6	0	1	0.964	198	0.439083
	1104	E	0	0	6	0	0	0	0	8	0	0.969	190	0.446911
	1137	E	8	8	0	8	8	8	8	3	8	0.975	289	0.630994

[†] T (for training), E (for evaluating)

Table 5. Results of experiments for best AI models of flow lines using 800 training data

Activation Function	Number of Layers	Batch Size	Test Data Set		R^2 (%)	Relative Errors (%)			
			Type	Num		Average	Max	Min	Range
Softmax	4	8	Original	200	99.52	-0.1342	2.6365	-1.8019	4.4384
			Extreme	100	99.09	0.0125	5.9161	-2.1305	8.0466
Softplus	7	8	Original	200	99.01	-0.0060	4.2971	-1.9345	6.2316
			Extreme	100	97.33	0.2457	7.5908	-3.3077	10.8984
ELU	4	8	Original	200	97.87	-0.2693	8.4566	-5.8605	14.3171
			Extreme	100	96.42	-0.2171	4.3383	-8.4497	12.7880
Sigmoid	8	8	Original	200	96.27	-0.4378	4.8065	-5.1279	9.9345
			Extreme	100	94.49	-0.3400	7.3903	-5.7963	13.1867
SoftSign	3	16	Original	200	95.78	0.0201	5.8027	-4.8042	10.6069
			Extreme	100	92.36	-0.0988	7.9637	-9.0377	17.0014
ReLU	3	16	Original	200	93.06	0.2117	8.1961	-10.2629	18.4591
			Extreme	100	77.48	1.6627	15.9199	-16.6169	32.5367

* Optimal values of parameters are determined based on 200 original test data

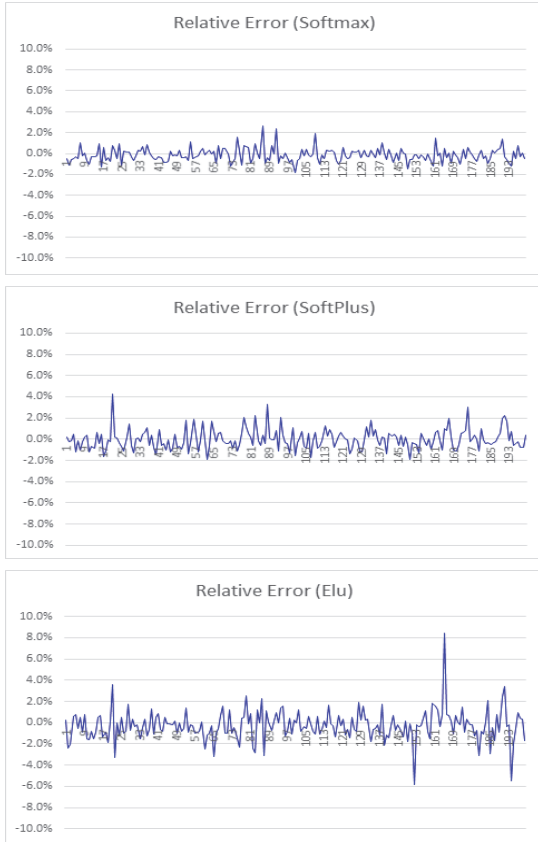


Fig. 7. Relative errors of 200 original test data when best DNN model is selected for each activation function

Figure 7은 평가용 최초데이터 200개에 대해 활성화 함수를 Softmax, Softplus, ELU로 했을 때 Table 5에 제시된 상대오차를 시각화한 것이다. 이 그래프에서 활성화 함수를 Softmax로 했을 때 전반적으로 상대오차도 작고,

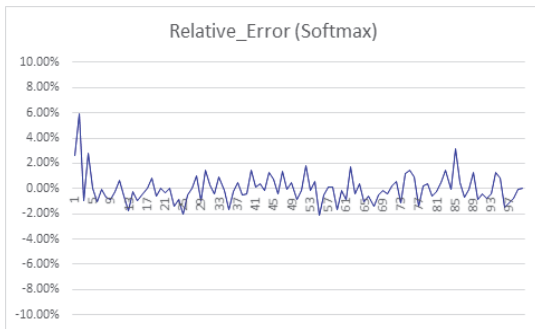
상대오차의 범위도 가장 적다는 것을 쉽게 알 수 있다. 따라서 Softmax를 이용한 학습모형이 가장 안정적으로 작동하고 있다고 결론을 내릴 수 있다.

다음으로 800개의 최초데이터를 이용하여 학습이 완료된 최초모델(Figure 6에서 First Model로 표현)에 100개의 평가용 극단데이터를 적용하였을 때 정확도가 얼마나 떨어지는지를 실험하였다. 이 비교 결과도 Table 5에 같이 제시되어 있다. 분석 결과 극단데이터에 대해 모든 경우 R^2 값이 하락하였고, 상대오차의 범위도 크게 증가하였음을 알 수 있다. 예외적으로 ELU 함수의 경우 R^2 값은 다소 감소하였지만 상대오차의 경우는 다소 개선된 것으로 나타났다. 하지만 Softmax를 사용한 경우보다는 모든 면에서 좋지 않은 결과를 보여주었다.

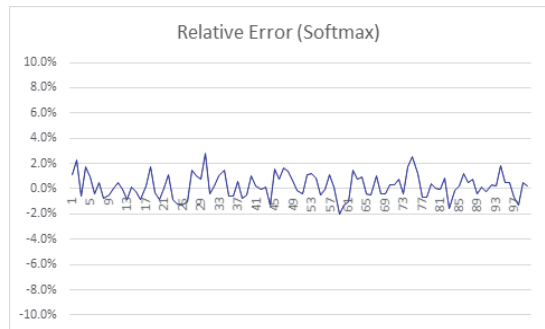
4.1.3 2차 실험

이와 같이 극단데이터의 정확성 문제를 해결하기 위해 Table 3을 이용하여 생성한 100개의 극단데이터를 최초 800개의 학습데이터에 추가하여 총 900개의 통합데이터로 다시 학습을 시켰다. 이 실험은 두 단계로 진행이 되었는데, 일차적으로 은닉층의 수, 벡치의 수는 새로 최적화를 수행하지 않고, 최초모델에서 최적화되었던 값을 그대로 사용하였으며(이를 신규모델(New Model)이라 부른다), 이차적으로 은닉층의 수와 벡치의 수를 평가데이터별 유형별로 다시 최적화시켰다(이를 신규최적화모델(New Optimized Model)이라 부른다).

학습된 모형을 이용하여 최초평가데이터, 극단평가데이터, 통합평가데이터를 각각 평가한 결과가 Table 6에 제시되어 있다. Softmax의 경우 최초모델보다 신규모델의 최초평가데이터의 R^2 값은 각각 99.52%와 99.53%로 대등소이했지만 극단평가데이터의 R^2 값은 99.09%에서



(a) Original DNN model



(b) New trained DNN model

Fig. 8. Relative errors of 100 extreme test data set in a serial line

Table 6. Results of experiments for best DNN models including extreme data set in serial lines

Activation Function	Number of Layers	Batch Size	Test Data Set		R^2	Relative Errors (%)			
			Type	Num.		Average	Max	Min	Range
Softmax	4 ⁺	8 ⁺	Original	200	99.53%	0.0555	4.1581	-1.4835	5.6416
			Extreme	100	99.22%	0.2327	2.8031	-2.0077	4.8108
			Total	300	99.48%	0.1146	4.1581	-2.0077	6.1658
	5	8	Original	200	99.57%	-0.1203	3.1503	-1.3611	4.5114
	5	8	Extreme	100	99.31%	0.1143	3.1220	-1.6741	4.7961
	5	8	Total	300	99.53%	-0.0421	3.1503	-1.6741	4.8244
Softplus	7 ⁺	8 ⁺	Original	200	95.78%	0.0890	6.6725	-4.2760	10.9486
			Extreme	100	93.02%	-0.3198	8.1887	-9.5910	17.7797
			Total	300	95.31%	-0.0473	8.1887	-9.5910	17.7797
	3	8	Original	200	98.91%	0.0892	4.0251	-2.9475	6.9726
	4	8	Extreme	100	97.91%	-0.0455	3.8761	-5.0114	8.8875
	3	8	Total	300	98.61%	0.0388	4.8968	-7.9583	12.8551
ELU	4 ⁺	8 ⁺	Original	200	98.65%	0.0009	3.3249	-3.5158	6.8406
			Extreme	100	97.88%	0.0321	4.1700	-5.3340	9.5040
			Total	300	98.54%	0.0113	4.1700	-5.3340	9.5040
	5	8	Original	200	98.70%	0.1553	3.7157	-4.1085	7.8242
	4	8	Extreme	100	97.54%	-0.1493	3.7939	-9.0318	12.8256
	4	8	Total	300	98.45%	-0.1590	3.7939	-9.0318	12.8256
Sigmoid	8 ⁺	8 ⁺	Original	200	97.54%	0.1807	5.1702	-4.1279	9.2981
			Extreme	100	98.07%	0.3103	4.8424	-2.6803	7.5228
			Total	300	97.98%	0.2239	5.1702	-4.1279	9.2981
Softsign	3 ⁺	16 ⁺	Original	200	96.63%	-0.0545	5.0990	-3.8013	8.9003
			Extreme	100	94.75%	0.0290	5.7937	-7.7195	13.5132
			Total	300	96.36%	-0.0267	5.7937	-7.7195	13.5132
ReLU	3 ⁺	16 ⁺	Original	200	92.99%	-0.4985	6.9695	-17.0421	24.0116
			Extreme	100	91.88%	-0.4699	6.6724	-8.7538	15.4261
			Total	300	93.36%	-0.4890	6.9695	-17.0421	24.0116

⁺ The optimal values of parameters are determined in Table 5

99.22%로 다소 높아졌다. 상대오차의 범위는 최초평가 데이터의 경우 다소 증가한 반면, 극단평가데이터의 경우는 8.0466에서 4.8108로 줄어들었다. Figure 8은 극단평가데이터 100개에 대해 최초학습모델과 신규학습모델의 상대오차를 보여준다.

반면에 Softplus의 경우 특이하게 신규모델이 두 종류의 데이터 세트에 대해 모두 나쁜 결과를 보여주었다. ELU의 경우 신규모델이 두 종류 데이터 세트에서 모두 개선된 결과를 보여주었다.

이 과정에서 학습모델의 모수값을 다시 최적화한다면

정확도가 얼마나 개선되는지 확인하기 위하여 900개의 데이터로 학습을 시킨 후 최적의 모수값 조합을 다시 찾아보았다. 그 결과도 Table 6에 같이 제시되어 있다. Softmax의 경우에는 3종류 데이터 세트에 대해 은닉층의 수가 4에서 5로 바뀌었는데 그 결과 R^2 값도 다소 증가하였다. 반면에 Softplus의 경우에는 당초 최적의 은닉층의 수가 7이었지만 다시 최적화를 한 결과 3 혹은 4의 값이 되었으며, 이 경우에 정확도도 많이 향상되었다. 참고로 Table 5에 있는 최초모델 분석결과에서 Softplus의 최적 은닉층 수가 7일 때 R^2 값은 99.01%였는데 최초모델에

서 은닉층 수가 3 혹은 4일 때 R^2 값은 98.57%, 98.48%였다. ELU의 경우는 극단데이터와 통합데이터는 은닉층의 수가 4로 동일했던 반면 최초데이터는 4에서 5로 변경되었다. 하지만 정확도 측면에서는 큰 차이가 없었다.

Figure 9는 극단평가데이터 100개에 대해 Softmax를 사용했을 때 당초 학습모형과 추가 학습모형에서 생성된 학습률의 상대오차를 시각화한 것이다. 이 그래프에서 극단데이터를 추가하여 새로 학습시킨 모델의 정확도가 개선된다는 것을 알 수 있다.

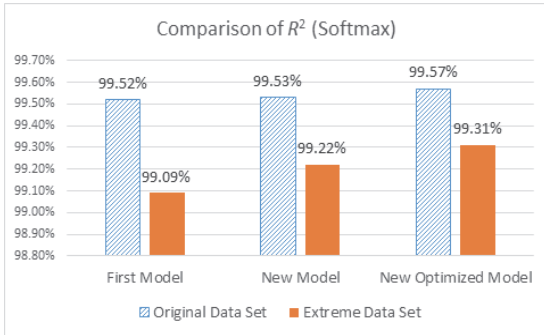


Fig. 9. Improvement of R^2 for 100 extreme test data set

4.2 조립라인

앞서 직렬라인에서 분석한 내용들이 다른 형태의 제조 시스템에도 유사하게 적용되는지 확인하기 위해 조립라인에도 동일한 방법으로 실험을 수행하였다. 본 연구에서 고려하는 조립라인은 Figure 3과 같이 18개의 공정으로 공정간 버퍼 17곳에 대한 변수와 MTTF, MTTR 관련 변수 2개 등 총 19개의 변수로 구성되어 있다. 각 변수가 가지는 값의 범위는 부분라인 내부의 버퍼($x_1 \sim x_5, x_7 \sim x_{12}, x_{13} \sim x_{17}$)은 0~5, 조립을 위한 버퍼(x_6, x_{12})의 경우는 1~10의 범위로 제한하였다. MTTF, MTTR 관련 변수의 범위는 직렬라인과 동일하게 적용하였다. 전체적인 분석 절차는 Figure 6에 제시된 절차와 동일하다.

4.2.1 실험설계

최적의 학습모델을 얻기 위해 Table 7과 같이 모수들의 종류를 정하였는데 직렬라인에서는 노드의 수를 한가지로 고정하였지만 조립라인에서는 4가지의 경우를 고려하였다. 따라서 총 실험 조합의 수는 $4 \times 10 \times 5 \times 7 = 1400$ 이다.

Table 7. Combinations of experiments for assembly lines

Parameters	Values
Number of Nodes	19, 38, 57, 79
Number of Epochs	2500
Number of Layers	1, 2, 3, .. 9, 10
Number of Batches	8, 16, 32, 64, 128
Activation Functions	Softmax, SoftSign, ELU, ReLU, Softplus, Sigmoid
Optimizer	ADAM

DNN 모델 결정을 위해 총 1200개의 데이터를 임의로 추출하여 최초데이터 세트를 만들었다. 그 중에서 900개의 데이터를 변수의 범위내에서 임의로 추출하여 학습에 사용하였으며, 나머지 300개의 데이터를 평가에 사용하였다. 다음으로 인공지능 모형에서는 추가 학습을 통하여 정확도를 높일 가능성이 있는지 확인하고자 추가로 극단데이터 세트를 마련하였다. 극단데이터 세트는 변수 ($x_1 \sim x_{17}$)의 생성확률을 Table 8과 같이 정하여 각 변수의 양측 극단 값이 많이 포함된 200개 생성하였다. 이중 임의로 100개는 평가용으로 하고, 나머지 100개는 차후에 최초 학습데이터 900개에 추가하여 1000개 데이터로 학습을 시키는데 활용하였다.

Table 8. Probability for generating extreme data set

Buffer Size ($x_1 \sim x_5, x_7 \sim x_{11}, x_{13} \sim x_{17}$)	0	1	2	3	4	5				
Probability(%)	40	10	5	5	5	35				
Buffer Size (x_6, x_{12})	1	2	3	4	5	6	7	8	9	10
Probability(%)	40	5	3	3	3	3	3	3	3	34

4.2.2 1차 실험

다음 Table 9는 각 활성화함수별로 200개의 조합에 대해 인공지능 모델을 학습시켜 평가데이터 300개에 대해 가장 우수한 조건을 찾아낸 결과다. 분석결과 활성화함수 측면에서는 흐름라인의 경우와 마찬가지로 Softmax나 Softplus가 우수한 성과를 보여주고 있다. R^2 의 경우 Softmax가 다소 우수하였지만 상대오차의 평균이나 범위 관점에서는 Softplus가 다소 우수한 것으로 분석되었다. 하지만 3.1절 직렬라인에 비해서는 R^2 값도 조금 낮고, 상대오차의 범위도 좀 더 큰 것으로 분석되었다. 그

Table 9. Results of experiments for best AI models of assembly lines

Activation Function	Number of Nodes	Number of Layers	Batch Size	Test Data Set		R^2	Relative Errors (%)			
				Type	Num.		Average	Max	Min	Range
Softmax	57	5	8	Original	300	96.84%	-0.3263	4.3806	-4.5756	8.9562
				Extreme	100	92.55%	0.2562	6.5368	-14.1883	20.7252
Softplus	57	4	8	Original	300	96.45%	0.0060	4.5771	-4.0472	8.6243
				Extreme	100	92.31%	0.6970	6.8463	-5.3219	12.1682
Sigmoid	76	6	16	Original	300	96.05%	-0.0494	5.5091	-4.7098	10.2189
				Extreme	100	92.52%	-0.3807	4.3578	-9.4977	13.8555
Elu	38	5	64	Original	300	95.42%	0.0937	5.7705	-5.2485	11.0190
				Extreme	100	88.13%	-0.4906	9.5862	-6.6056	16.1918
Softsign	57	4	64	Original	300	94.08%	0.0241	6.9721	-6.3705	13.3426
				Extreme	100	84.27%	1.0537	10.6865	-10.0207	20.7071
Relu	76	2	8	Original	300	91.02%	0.1904	8.3518	-10.9288	19.2806
				Extreme	100	70.81%	-0.8350	11.3354	-10.9762	22.3116

이유는 변수의 수가 11개에서 19개로 증가한 반면에 학습에 사용한 데이터의 수는 800개에서 900개로 100개만 증가하였기 때문으로 추측한다.

이와 같이 900개의 최초데이터를 이용하여 이미 학습이 완료된 모형으로 100개의 생성된 극단데이터 세트에 대해 적용하여 평가를 수행한 결과도 Table 9에 같이 제시하였다. 분석 결과 극단데이터세트에 대해 모든 경우 R^2 값이 4% 이상 하락하였고, 상대오차의 범위도 크게 증가한 것을 알 수 있다. 특히 Softmax의 경우 특정한 한 개의 데이터에 대해 상대오차가 -14.19% 발생하는 특이한 상황이 발생하였다(Figure 10a 참조).

4.2.3 2차 실험

이와 같이 극단 데이터의 정확성 문제를 해결하기 위해 Table 8을 이용하여 생성한 200개의 데이터중 100개를 선택하여 당초 900개의 학습데이터에 추가한 후 총 1000개의 데이터로 다시 학습을 시켰다. 흐름라인의 경우와 동일하게 일차적으로 노드의 개수, 은닉층의 개수, 배치의 개수는 새로 최적화를 수행하지 않고, 당초 학습모형에서 최적화 되었던 값을 그대로 사용하였다. 그 후 이차적으로 각 극단 데이터 세트 100개를 포함한 전체 400개 평가 데이터 세트에 대해 최적의 조건을 다시 찾아 분석한 비교결과를 Table 10에 제시하였다. 실제 실험은 더 많은 활성화 함수에 대해 수행하였지만 지면관계상 상위 4개의 활성화 함수에 대한 결과만 제

시하였다.

Softmax의 경우 극단데이터 100개에 대해 최초 학습 모델에서는 R^2 값이 92.55%였는데, 신규학습모델에서는 95.40%로 증가하였으며, 모수를 다시 최적화 했을 때는 96.45%로 증가하였다. 또한 상대오차의 범위도 20.73%대에서 추가학습을 했을 때 9.12%와 9.33%로 감소하였다.

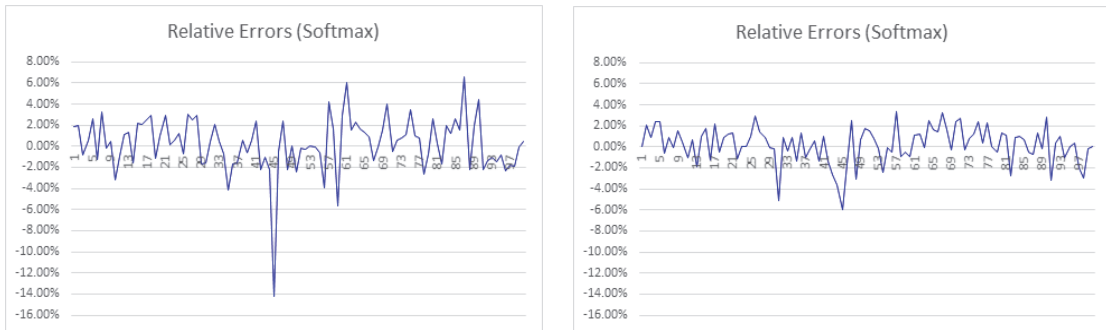
Figure 10(a)는 활성화함수를 Softmax로 했을 때, 최초학습모델을 이용하여 극단데이터 100개에 대해 상대오차를 평가한 그래프이며, Figure 10(b)는 극단데이터 100개를 포함한 1000개의 학습데이터로 다시 학습을 시키고, 극단 평가데이터 100개에 대해 모수를 다시 최적화시킨 후 상대오차를 평가한 그래프다. 이를 통하여 극단데이터를 학습에 일정부분 활용하는 것이 정확도를 높이는 방법이라는 것을 확인할 수 있었다. 하지만 상대오차의 범위가 여전히 만족할 수준은 아니라고 판단한다.

다른 활성화 함수의 경우에도 4.1절의 직렬라인에서 보여준 결과와 유사한 형태를 보여주었다. 본 연구에서 고려하는 직렬라인이나 조립라인의 경우 DNN 모델에 적합한 활성화함수는 Softmax, Softplus 등이며, Sigmoid나 ELU도 나쁘지 않은 결과를 제시해 주었다. 다만 활성화 함수 선택에서 조심해야 할 점은 Softmax의 경우 은닉층의 개수가 6~7개 이상이 되면 R^2 값이 이상치가 나오는 등 안정적인 상태를 보여주지 못한다는 점이다. 다른 활성화수들에서는 이런 현상이 발생하지 않았다.

Table 10. Results of experiments using new trained DNN models for assembly lines

Activation Function	Number of Nodes	Number of Layers	Batch Size	Test Data Set		R^2	Relative Errors (%)			
				Type	Num		Average	Max	Min	Range
Softmax	57 ⁺	5 ⁺	8 ⁺	Original	300	97.05%	0.0581	4.6768	-4.8614	9.5381
				Extreme	100	95.40%	-0.0746	3.4906	-5.6285	9.1191
				Total	400	97.13%	0.0250	4.6768	-5.6285	10.3053
	57	3	8	Original	300	97.81%	0.0019	4.0037	-4.0300	8.0337
	57	3	8	Extreme	100	96.45%	0.2231	3.3125	-6.0163	9.3288
	57	3	8	Total	400	97.84%	0.0572	4.0037	-6.0163	10.0201
Softplus	57 ⁺	4 ⁺	8 ⁺	Original	300	95.79%	0.0391	7.4454	-5.5018	12.9472
				Extreme	100	92.77%	-0.3761	7.0467	-6.6269	13.6736
				Total	400	95.75%	-0.0647	7.4454	-6.6269	14.0723
	38	4	8	Original	300	96.42%	-0.1090	4.9349	-5.2673	10.2022
	76	6	16	Extreme	100	95.65%	-0.1272	6.1907	-4.9271	11.1179
	76	3	16	Total	400	96.64%	-0.0859	5.9545	-5.6027	11.5572
Sigmoid	76 ⁺	6 ⁺	16 ⁺	Original	300	94.37%	-0.0557	5.6389	-7.6290	13.2679
				Extreme	100	94.81%	0.0847	5.9456	-6.1363	12.0818
				Total	400	95.29%	-0.0206	5.9456	-7.6290	13.5746
	38	7	8	Original	300	96.27%	-0.0231	7.0276	-5.2565	12.2842
	38	7	8	Extreme	100	95.92%	-0.0525	4.6496	-4.3828	9.0325
	38	7	8	Total	400	96.74%	-0.0304	7.0276	-5.2565	12.2842
Elu	38 ⁺	5 ⁺	64 ⁺	Original	300	94.96%	-0.0886	6.1003	-5.2330	11.3332
				Extreme	100	93.71%	-0.1140	4.8184	-5.4286	10.2470
				Total	400	95.43%	-0.0949	6.1003	-5.4286	11.5288
	57	7	128	Original	300	95.87%	-0.0137	5.1395	-4.9344	10.0738
	57	9	64	Extreme	100	95.06%	-0.3148	5.5084	-5.2765	10.7849
	57	9	128	Total	400	96.27%	0.0925	5.4900	-5.3617	10.8516

⁺ The optimal values of parameters are determined in Table 9



(a) Original DNN model

(b) New optimized DNN model

Fig. 10. Relative errors of 100 extreme test data set in an assembly line

4.3 특수한 형태의 조립라인 (차체공장)

앞서 분석한 직렬라인과 조립라인에서 학습모델이 보여준 결과의 행태는 대등소이하였다. 일반적으로 AI모델은 많은 양의 학습데이터가 필요하다고 한다. 따라서 이번 절에서는 “생산률을 추정하는데 구태여 AI 모델을 사용할 필요가 있을까?”에 대한 의문을 확인해 보고자 한다. 이를 위해 실험계획(DOE)를 활용하여 데이터 수를 최소화한 후 2차 회귀모형(Quadratic Regression Model)을 이용한 메타모델(Lee et al., 2020a)과 DNN모델의 성능을 비교하고자 한다. 대상 시스템은 2.3절에서 설명한 Layered Build 공법을 적용하고, 2차종 혼류생산을 하는 자동차 차체공장의 일부 라인으로 총 9개의 변수를 가지고 있다.

4.3.1 실험 설계

먼저 DNN 모형 학습을 위해 두 종류의 데이터를 마련하였다. 첫 번째 학습용 데이터는 Lee et al.(2020a)에서 메타모델 개발을 위해 사용한 156개의 데이터이고, 두 번째 학습용 데이터는 임의로 추출한 156개의 데이터다. 평가용으로도 두 종류의 데이터를 만들었는데, 첫 번째는 임의로 추출한 100개의 데이터다. 두 번째 평가데이터는 4.1절과 4.2절에서 수행했던 방법과 유사하게 각 변수값의 범위에서 양 극단의 값이 나올 확률을 대폭 높여서 얻은 극단적 데이터 100개를 정의하였다.

최적의 DNN 모형을 찾기 위해 설정한 모수들은 Table 11과 같은데, 앞서 직렬라인과 조립라인에서 수행했던 최적조합의 결과를 참고하여 그 범위를 다소 축소시켰다.

Table 11. Combinations of experiments for body shop

Parameters	Values
Number of Nodes	18, 27, 36
Number of Epochs	2500
Number of Layers	3, 4, 5, 6, 7
Number of Batches	8, 16, 32
Activation Functions	Softmax, Elu, Softplus, Sigmoid
Optimizer	ADAM

4.3.2 1차 실험

1차 실험의 목적은 기계학습을 할 때, 메타모델에서 사용하는 DOE 데이터와, 같은 수의 임의데이터 중 어느 것이 효과적인지 비교하는 것으로 그 결과는 Table 12에

제시되어 있다.

먼저 DOE 데이터를 이용하여 학습하고, 임의의 100개 데이터에 대해 평가를 한 경우 메타모델의 R^2 값은 99.71%였고, DNN 모델의 경우 활성화 함수에 따라 차이가 있지만 R^2 값이 99.30%~99.72% 수준이었다. 메타모델의 상대오차 범위는 12.1755%였으며, DNN 모델의 상대오차 범위는 활성화함수에 따라 다르지만 9.3075%~19.1844%였다(Figure 11 참조). 앞서 분석한 두 종류의 시스템에서 Softmax가 가장 우수했던 결과와는 달리, 세 번째 시스템에서는 활성화함수를 ELU로 하였을 때 가장 우수한 결과를 얻었다. ELU의 경우 상대오차의 범위도 메타모델에 비해 2.87% 줄어들었다.

다음으로 DNN 모델에서 156개의 DOE 데이터 대신에 임의 데이터 156개로 학습을 수행한 경우를 비교하였다. 그 결과, 예상밖으로 임의데이터를 이용해 학습을 수행한 DNN 모델의 R^2 값과 상대오차의 범위 모두 ELU를 제외하고는 모두 개선되었다. 메타모델과 비교할 경우, 네 종류의 활성화함수에 대해 R^2 값은 높은 경우와 낮은 경우가 공존하였지만, 상대오차의 범위는 모든 경우 개선되었다. 결론적으로 기계학습에는 DOE 데이터보다는 임의 데이터가 효과적이었다.

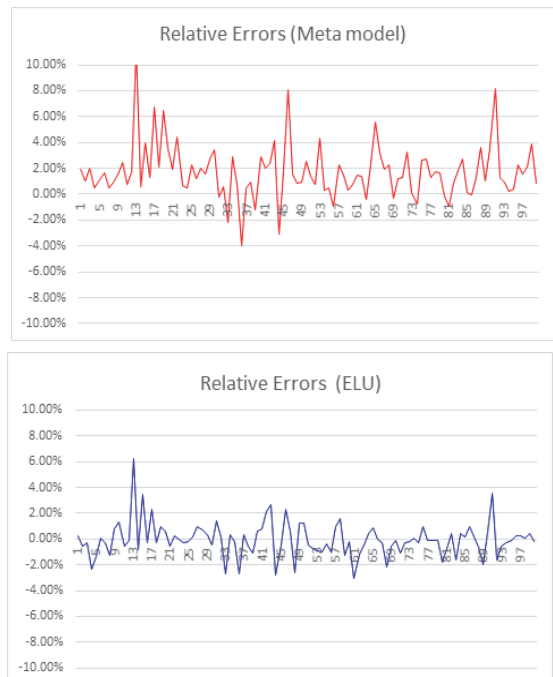


Fig. 11. Relative errors for random test data set in a body shop (training with DOE data set)

Table 12. Comparison between meta model and DNN model in a body shop

	Activation Function	(A)	(B)	(C)	Training Data		Test Data		R^2	Relative Errors (%)			
					Type	Num	Type	Num		Avg.	Max	Min	Range
Meta model (Lee et al. ^[10])					DOE	156	Random	100	99.71	-0.0115	7.0523	-5.1232	12.1755
					DOE	156	Extreme	100	95.92	0.2606	26.3472	-26.0145	52.3617
DNN	Softmax	27	5	16	DOE	156	Random	100	99.45	0.6855	7.7334	-4.7805	12.5139
	Softplus	18	3	8					99.55	-0.0970	7.6994	-7.8547	15.5542
	Sigmoid	36	4	8					99.30	0.0091	10.9908	-8.1936	19.1844
	ELU	36	3	16					99.72	-0.0397	6.2486	-3.0589	9.3075
	Softmax	27	4	8	Random	156			99.59	-0.1055	4.1124	-4.3595	8.4719
	Softplus	27	4	8					99.74	0.1358	6.1454	-3.2649	9.4102
	Sigmoid	36	5	8					99.57	-0.0510	4.3889	-4.2482	8.6372
	ELU	36	4	8					99.71	0.0220	5.6958	-4.3464	10.0422
	Softmax	18	3	8	DOE	156	Extreme	100	95.17	5.9137	40.3824	-9.2407	49.6231
	Softplus	36	4	8					96.16	3.5233	32.3560	-12.5046	44.8606
	Sigmoid	36	4	8					95.24	4.4252	39.4433	-13.0658	52.5091
	ELU	18	5	16					96.41	4.5929	27.7563	-10.9040	38.6604
	Softmax	27	4	8	Random	156			96.43	1.8790	27.3550	-15.9411	43.2961
	Softplus	27	5	8					96.86	0.9975	24.9034	-13.3975	38.3009
	Sigmoid	27	5	8					96.38	1.5285	26.8025	-12.4094	39.2119
	ELU	36	4	8					97.13	1.1120	25.9412	-13.1346	39.0758

※ (A) : Number of nodes, (B) : Number of layers, (C) : Batch size

4.3.3 2차 실험

1차 실험에서 메타모델이나 DNN 모델이나 156개의 소수 데이터를 이용하여 임의의 평가용 데이터를 분석해 본 결과 의미있는 정확성을 얻을 수 있다는 것을 확인하였다. 하지만 당초 본 연구에서 의문을 가졌던 극단적인 데이터에 대해 개발한 모델들이 얼마나 예측을 잘하는가를 확인할 필요가 있다. 따라서 4.3.1절에서 설명한 바와 같이 극단적 데이터 100개를 선정하였다. 이 데이터를 평가에 사용해 얻은 분석결과도 Table 12에 제시되어 있다.

메타모델을 사용한 경우 극단적 평가데이터의 R^2 값은 95.92%, 상대오차의 범위는 52.36%였는데, 이는 DOE 데이터를 이용해 학습을 수행한 DNN 모델의 결과와 비슷한 수준이었다. 반면에 임의의 데이터를 이용해 학습을 수행한 DNN 모델의 경우 활성화 함수가 ELU일때 R^2 값은 97.13%, 상대오차의 범위는 39.08%로 임의의 데이터를 이용하여 학습을 시킨 경우가 오히려 좋은 결과를 보여주었다. 하지만 극단적 데이터에 대한 예측 수준은 메타 모델이나 DNN 모델 모두 최적화문제에 사용할 정도의 정확성을 확보하지 못했다고 하겠다. 따라서 앞의 4.1.3절, 4.2.3절에 제시된 바와 같이 학습에 극단적 데이터를

추가하여 학습모델을 만들면 예측의 정확성이 향상될 것이다. Figure 12는 1, 2차 실험에서 두 종류의 학습데이터와 두 종류의 평가데이터를 이용하여 개발한 DNN 모델의 R^2 값을 종합한 것이다.

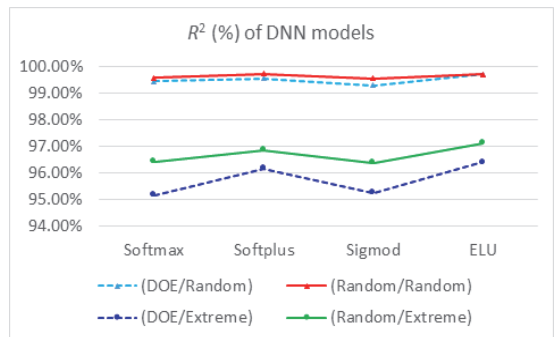


Fig. 12. R^2 of DNN models with different data sets

마지막으로 학습데이터 수와 평가데이터 수를 어떻게 배분하는 것이 효과적인지 비교하여 보았다. 비교는 Lee et al.(2020a)에서 사용하였던 2차회귀분석(Quadratic Regression) 방법과 활성화 함수를 ELU로 한 DNN 모델

을 선택하였다. 이때, DNN 모델의 노드 수 등 모수들은 최적화시키지 않고 특정한 값으로 고정하여 비교하였다.

이를 위해 별도의 1221개 임의데이터를 추출하여 일부는 회귀모형 개발 (또는 DNN 모델 학습)에 사용하고 일부는 평가에 사용하였다. 예를 들어(500/721)은 1221개 데이터 중 임의로 500개를 모델개발에 사용하고, 나머지 721개를 평가에 사용하였다는 의미다. 결과적으로 모델 개발에 사용한 데이터의 수가 늘어날수록 R^2 값이 증가하였으며, DNN 모델이 2차회귀모형보다 좀 더 우수한 결과를 보여주었다(Figure 13 참조).

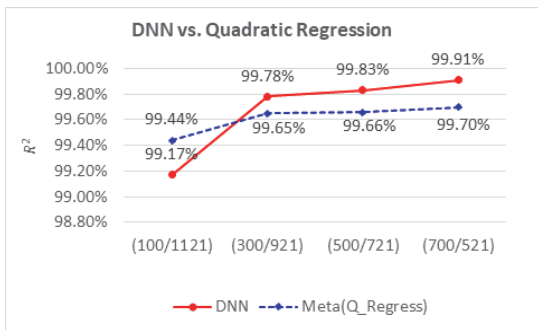


Fig. 13. Comparing accuracies between DNN and Quadratic Regression

5. 결론

제조시스템을 최적화하기 위해서는 의사결정변수 및 통제변수의 값이 변할 때 시스템의 성능을 예측해야 한다. 이때, 정확한 수학적 모델을 개발할 수 있다면 가장 이상적이겠지만 현실적으로 불가능하다. 따라서 대안으로 수학적 근사해법, 시뮬레이션, 수리모형을 이용한 메타모델, 인공지능모델 등을 사용할 수밖에 없다.

본 논문에서는 직렬라인, 조립라인, 자동차 차체공정의 일부인 특수한 조립라인 등에 대해 학습모델의 하나인 DNN 모델의 활용가능성을 다양한 실험을 통하여 분석하였다. 이쉽게도 극단적 데이터에 대해서 정확도가 떨어진다라는 점은 수학적 근사해법, 메타모델, DNN 모델 모두 어느 정도의 한계를 가지고 있었다.

본 연구를 통하여 확인한 주요 사항은 다음과 같다. 첫째, 제조시스템의 생산률을 이용한 최적화 문제를 다루기 위해 생산률을 추정하는 방법으로 지도학습을 이용한 DNN 모델은 충분히 활용가능하다. 둘째, 일반적으로 학습을 위해서는 많은 데이터가 필요한데 의사결정변수 및 통제변수의 수에 따라 다르기는 하지만 수백개에서 1~2

천개의 데이터로 충분하다. 셋째, 일반적으로 메타모델을 개발할 필요한 데이터 수를 최소화시키기 위해 DOE 방법을 사용하는데 학습모델에서는 동수의 임의데이터를 사용하는 것이 오히려 낫다. 넷째, 학습을 위해 일반적으로 임의로 데이터를 선택하는데, 제조시스템 설계문제의 특성상 변수의 범위의 최대, 최소값이 많이 포함되는 극단적 데이터를 학습데이터에 포함시키는 것이 추정의 정확도를 높인다. 다섯째, DNN 모델을 활용하기 위해서 다양한 활성화함수를 사용할 수 있는데, Softmax나 ELU 등이 좋은 결과를 보여주었다. 다만 Softmax를 사용할 때는 은닉층의 수를 크게 하면 안된다.

특정 시스템에 대해 학습이 완료된 DNN 모델을 이용하면 동일 시스템에서 각종 의사결정변수나 매개변수의 값이 정의된 범위 내에서 변경될 때 새롭게 시뮬레이션을 수행할 필요가 없다. 단지 변경된 변수값만 DNN 모델에 입력함으로써 생산률을 쉽게 추정할 수 있다. 따라서 BAP 문제같은 최적화 문제의 최적화 과정에 효과적으로 사용할 수 있다.

향후 연구주제로는 개발된 DNN 모델을 이용하여 최적화 문제를 해결할 때 발생하는 문제점을 분석하는 것, 제조시스템 효율의 특성을 잘 반영할 수 있는 활성화 함수를 개발하는 것과, 임의의 학습데이터와 극단적 학습데이터를 어떻게 혼합하여 학습을 시키는 것이 효율적인지에 대한 체계적인 틀을 개발하는 것이다.

References

1. Apicella, A., Donnarumma, F., Isgrò, F. and Prevete, R., “A Survey on Modern Trainable Activation Functions”, *Neural Networks*, Vol. 138, pp. 14-32, 2021.
2. Can, B. and Heavey, C., “Comparison of Experimental Designs for Simulation-based Symbolic Regression of Manufacturing Systems”, *Computers & Industrial Engineering*, Vol. 61, No. 3, pp. 447-462, 2011.
3. Can, B. and Heavey, C., “A Comparison of Genetic Programming and Artificial Neural Networks in Metamodeling of Discrete-event Simulation Models”, *Computers & Operations Research*, Vol. 39, No. 2, pp.424-436, 2012.
4. Dubey, S. R., Singh, S. K. and Chaudhuri, B. B., “Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark”, arXiv:2109.

- 14545v3 [cs.LG], 2022.
5. Fonseca, J. and Navarrese, D., “Artificial Neural Networks for Job Shop Simulation”, *Advanced Engineering Informatics*, Vol. 16, No. 4, pp. 241-246, 2002.
 6. Gershwin, S. B., *Manufacturing Systems Engineering*, Prentice-Hall, New Jersey, U.S.A., 1994.
 7. Kang, Z., Catal, C. and Tekinerdogan, B., “Machine Learning Applications in Production Lines: A Systematic Literature Review”, *Computers & Industrial Engineering*, Vol. 149, 106773, 2020.
 8. Khayyati, S. and Tan, B., “A Machine Learning Approach for Implementing Data-driven Production Control Policies,” *International Journal of Production Research*, Vol. 60, No. 10, pp. 3107-3128, 2022a.
 9. Khayyati, S. and Tan, B., “Supervised Learning Based Approximation Method for Single-Server Open Queuing Networks with Correlated Interarrival and Service Times”, *International Journal of Production Research*, Vol. 60, No. 22, pp.6822-6847, 2022b.
 10. Koyuncuoğlu, M. U., “Most Important Performance Evaluation Methods of Production Lines: A Comprehensive Review on Historical Perspective and Emerging Trends”, *Computers & Industrial Engineering*, Vol. 197, 110623, 2024.
 11. Lee, Y. H., Kim, D. O., Baek, K. M., Shin, Y. W. and Moon, D. H., “Optimizing Design Problem in an Automotive Body Assembly Line Considering Cost Factors”, *Journal of the Korea Society for Simulation*, Vol.29, No.4, pp.95-109, 2020a.
(이영훈, 김동욱, 백경민, 신양우, 문덕희, “비용요소를 고려한 자동차 차체조립라인의 설계 최적화”, *한국시물레이션학회논문지*, Vol.29, No.4, pp.95-109, 2020a.)
 12. Lee, Y.H., Shin, Y.W. and Moon, D.H., “Buffer Allocation Problem Using Meta-Model in an Automotive Body Shops with Mixed-model Production”, *Journal of the Korean Institute of Industrial Engineers*, Vol. 46, No. 3, pp. 296-308, 2020b.
(이영훈, 신양우, 문덕희 “혼류생산을 하는 자동차 차체공장에서 메타모델을 이용한 버퍼할당”, *대한산업공학회지*, Vol. 46, No. 3, pp. 296-308, 2020b.)
 13. Li, J. S. and Meerkov, S. M., *Production Systems Engineering*, Springer, New York, U.S.A., 2009.
 14. Moon, D. H., Kim, D. O. and Shin, Y. W., “Automotive Body Shop Design Problems Using Meta-models Considering Product-mix Change and Reconfiguration Strategy”, *Applied Sciences-Basel*, Vol.11, No.6, 2748, pp.1-20, 2021.
 15. Moon, D. H., Seo, Y. R., Kim, D. O., Shin, Y. W., and Ko. C. S., “ANN Meta-model based on Simulation for Estimating Production Rate in an Assembly Manufacturing System”, *Abstract Book of 32th European Conference on Operational Research (EURO 2022)*. pp.272, Aalto University, Espoo, Finland, 2022.
 16. Nwankpa, C. E., Ijomah, W., Gachagan, A. and Marshall, S., “Activation Functions: Comparison of Trends in Practice and Research for Deep Learning”, arXiv:1811.03378v1 [cs.LG], 8 Nov 2018.
 17. Papadopoulos, C. T., Li, J. and O’Kelly, M. E. J., “A Classification and Review of Timed Markov Models of Manufacturing Systems”, *Computers & Industrial Engineering*, Vol. 128, pp. 219-244, 2019.
 18. Ramachandran, P., Zoph, B. and Le, Q. V., “Searching for Activation Functions”, arXiv:1710.05941 [cs.NE], 2017.
 19. Shin, Y. W. and Moon, D. H., “Throughput of Flow Lines with Unreliable Parallel-machine Workstations and Blocking”, *Journal of Industrial Management and Optimization*, Vol.13, No.2, pp.901-916, 2017.



문 덕 희 (ORCID : <https://orcid.org/0000-0001-7660-4976> / dhmoon@changwon.ac.kr)

1984 한양대학교 산업공학과 공학사
1986 한국과학기술원 산업공학과 공학석사
1991 한국과학기술원 산업공학과 공학박사
1990~ 현재 창원대학교 산업시스템공학과 교수

관심분야 : Facilities Planning, 시뮬레이션 응용, Scheduling



핀 권 안 (ORCID : <https://orcid.org/0009-0003-9224-0684> / annie.tqa@gmail.com)

2015~ 2018 Viet Nam National University, Ho Chi Minh City, Faculty of Economics 학사
2019~ 2022 Worked in Operations Management at DB Schenker Vietnam
2023~ 2023 Worked in Business Development at YCH Group Vietnam
2024~ 현재 창원대학교 대학원 스마트제조융합협동과정 석사과정

관심분야 : Operations Management, SCM, Application of Machine Learning



김 동 옥 (ORCID : <https://orcid.org/0000-0002-7239-2002> / positive_ok@naver.com)

2013~ 2019 창원대학교 산업시스템공학과 공학사
2019~ 2021 창원대학교 친환경해양플랜트FEED과정 공학석사
2021~ 2022 창원대학교 산업기술연구원 연구원
2022~ 현재 현대자동차 생산정원개선팀

관심분야 : 생산시스템 분석, 시뮬레이션 응용, Scheduling



신 양 우 (ORCID : <https://orcid.org/0000-0002-3107-4569> / ywshin@changwon.ac.kr)

1984 경북대학교 수학과 이학사
1986 한국과학기술원 응용수학과 이학석사
1991 한국과학기술원 수학과 이학박사
1991~ 현재 창원대학교 통계학과 교수

관심분야 : 확률과정론, 대기행렬망이론, Manufacturing System Design