

블록 장난감 제조 공정 최적화 AI 경진대회

DACON | LG AI 해커톤

팀명_ 대왕이

팀원_ 손주호

01

문제소개 및 분석방향

02

탐색적 자료분석

03

모델 소개

04

독창성 & 확장성



01

대회문제 및 분석방향



AI 활용 블록 장난감 제조 공정 최적화

1. 어린이날 전후로 늘어난 블록 장난감 수요
2. 적기/적량 생산을 위한 공정 설계 알고리즘 개발



제약 사항 파악

원형 제작, 성형, 자르기 공정의 제약사항 파악



목적 함수 설정

목적 함수(손실 함수)의 수학적 모델링



모델 구축

신경망과 유전 알고리즘을 사용한 모델 구축



02

탐색적 자료분석

stock

- 4월1일 기준 재고

원형 제작



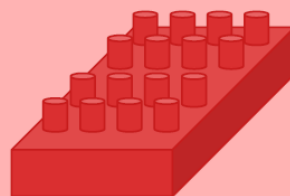
ID: PRT_1 ~ 4

- 일별 최대 투입 개수: 500개

- 원형제작공정 양품률: 98.5%



성형



ID: MOL_1 ~ 4

max_count

- 일별 최대 투입 개수

change_time

- 변경 소요 시간

- 성형공정 양품률: 97.5%



자르기



ID: BLK_1 ~ 4

cut_yield

- 자르기공정 월별 양품률

order

- BLK1~4의 일별 수요

stock

- 4월1일 기준 재고

	PRT_1	PRT_2	PRT_3	PRT_4	MOL_1	MOL_2	MOL_3	MOL_4	BLK_1	BLK_2	BLK_3	BLK_4
2020-04-01	0	258	0	0	1086	0	0	0	61158	87279	0	0



생산

order

- BLK1~4의 일별 수요

time	BLK_1	BLK_2	BLK_3	BLK_4
2020-06-16	31874	0	62911	0
2020-06-17	21250	0	31456	41395
2020-06-18	21250	0	31456	41395
2020-06-19	42499	0	62911	41395
2020-06-20	42499	0	41941	41395
2020-06-21	42499	0	41941	41395
2020-06-22	42499	0	41941	62092
2020-06-23	42499	0	62911	0
2020-06-24	21250	0	31456	62092
2020-06-25	21250	0	31456	62092
2020-06-26	42499	0	62911	62092
2020-06-27	42499	0	41941	62092
2020-06-28	42499	0	41941	62092
2020-06-29	42499	0	41941	31046
2020-06-30	42499	0	62911	0

max_count

- “성형” 공정
- 일별 최대 투입 개수

	개수
4.1 ~ 4.14	0
4.15 ~ 4.30	140.590741
5.1 ~ 5.31	140.805556
6.1 ~ 6.30	141.018518

change_time

- “성형” 공정
- 변경 소요 시간

	시간
MOL_1 ↔ MOL_2	6
MOL_3 ↔ MOL_4	6
etc	13

cut_yield

- “자르기” 공정
- 월별 양품률

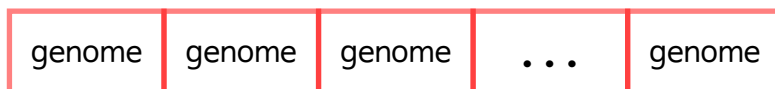
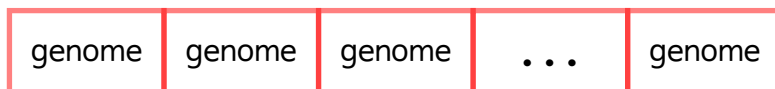
	20년 4월	20년 5월	20년 6월
BLK_1	85.1	85.1	85.1
BLK_2	90.1	90.1	90.1
BLK_3	71.0	74.2	75.9
BLK_4	70.0	73.2	74.9



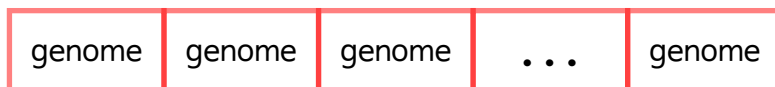
03

모델 소개

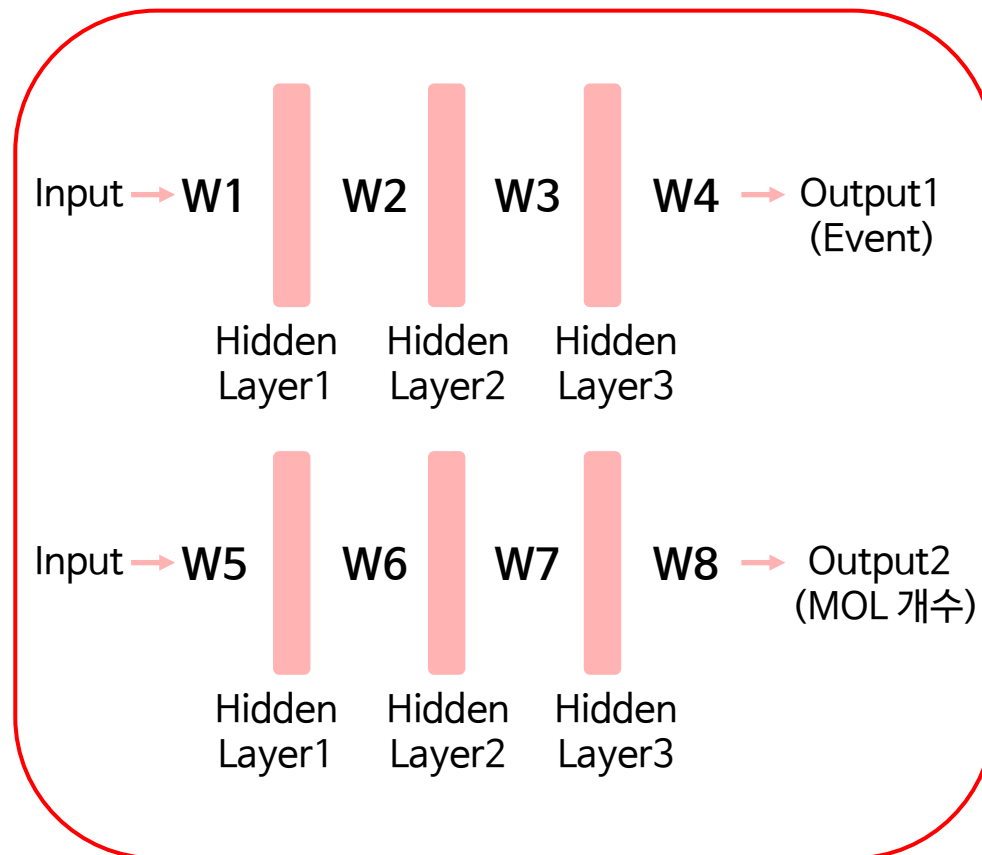
1 세대



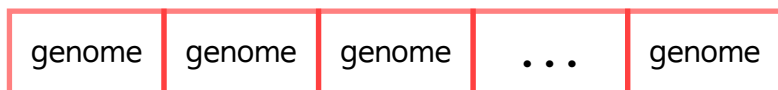
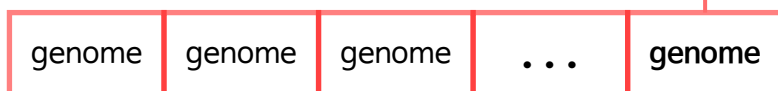
2 세대



Genome



1 세대



2 세대



Genome

mask
updateNeural
Networkmask
update

score

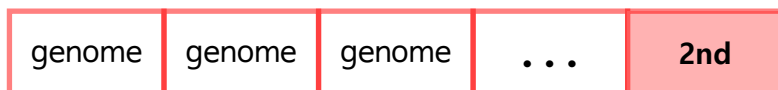
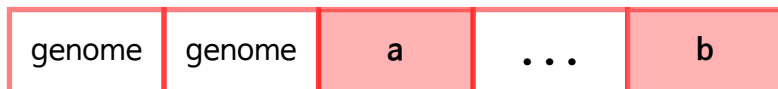
Simulator

생성 BLK 개수
= (MOL 개수 + 재고 개수) * 양품률
필요 BLK 개수
= order 개수

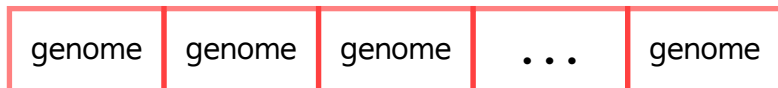
score

= 생성 BLK 개수 - 필요 BLK 개수

1 세대



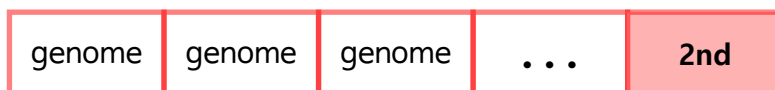
2 세대



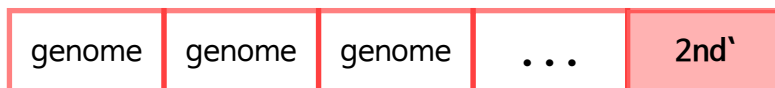
Genome

$$W_{2nd} = [W_a \quad W_b]$$

1 세대



2 세대



Genome

$$\begin{aligned}
 W_{2nd'} &= \\
 W_{2nd} & * \\
 [normal] & * \\
 [0 \sim 2 \text{ int}] &
 \end{aligned}$$

분석 환경

CPU : Intel(R) Xeon(R) CPU E5-2630
(2.30GHz, 24core)

Memory : 65,943,312kb (64GB)

pandas = 1.0.5

numpy = 1.18.5

matplotlib = 3.2.2

python = 3.8.3

	Hyper parameter
세대당 유전자 수	50
베스트 유전자 수	5
자식 유전자 생성 수	5
돌연변이 비율	50%
세대 수	160



	시간	점수
훈련: 3개월	7,247초 (2시간)	-
예측	5초	87.96103

$$\text{점수} = 50F(p, 10N) + 20F(q, 10N) + \frac{20F(c, M)}{(1+0.1c_n)} + \frac{20F(s, M)}{(1+0.1s_n)}, \text{ which } F(x, a) = \begin{cases} 1 - \frac{x}{a} & (\text{if } x < a) \\ 0 & (\text{else}) \end{cases}$$



분기 단위 주문 파악

3개월 단위의 주문 수량을 파악



분기마다 모델 학습

모델의 학습에 2시간 소요되므로, 빠르게 다음 분기의 공정 설계 가능



공정 최적화

예측한 수량대로 라인별 블록 생산

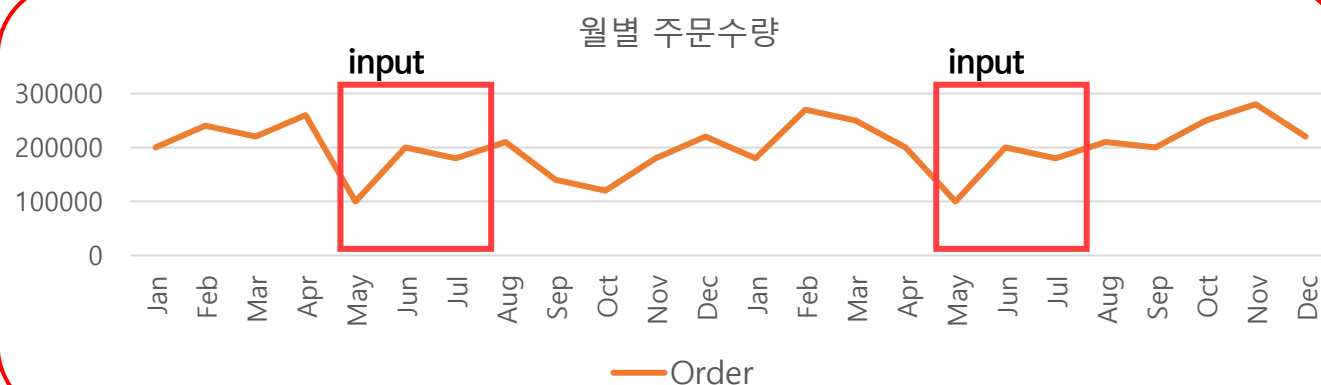
분기별 n세대 유전자 보관

n 세대

genome	genome	genome	...	genome
--------	--------	--------	-----	--------



비슷한 추이를 보이는 분기



사용했던 유전자의 재사용
기존 유전자 + 새로운 input으로 학습

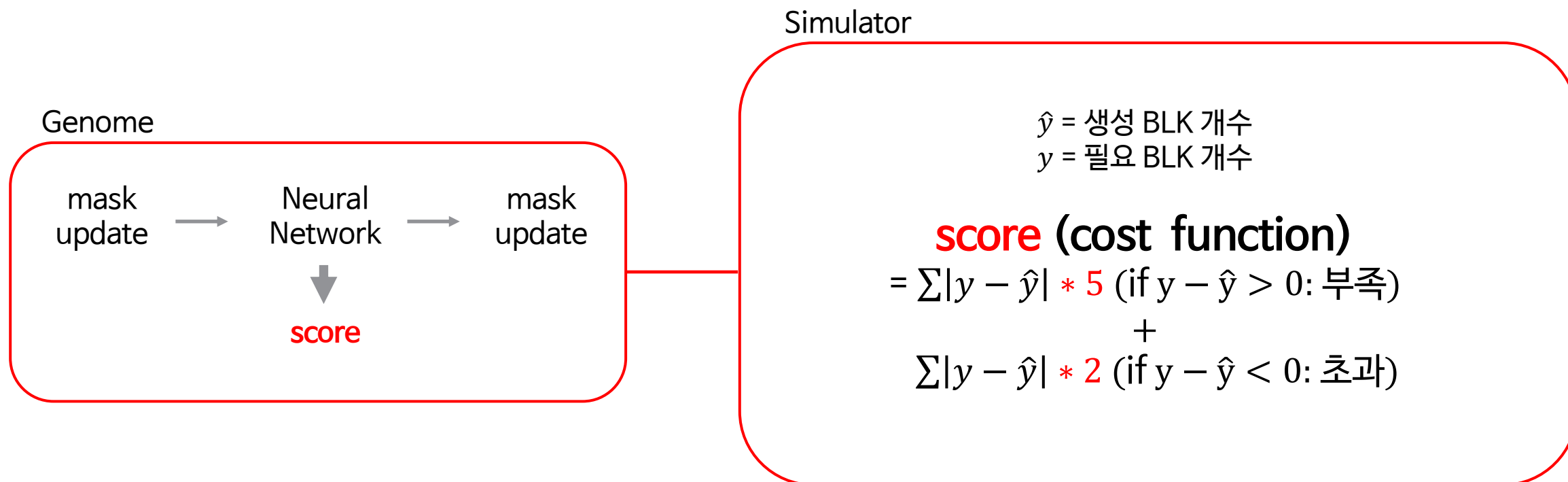


학습 EPOCH (반복 회수) 감소,
효율성 제고



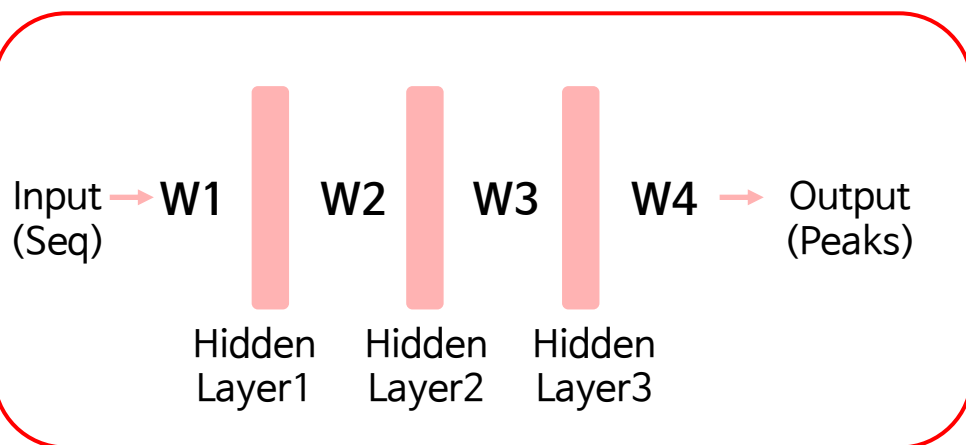
04

독창성 & 확장성



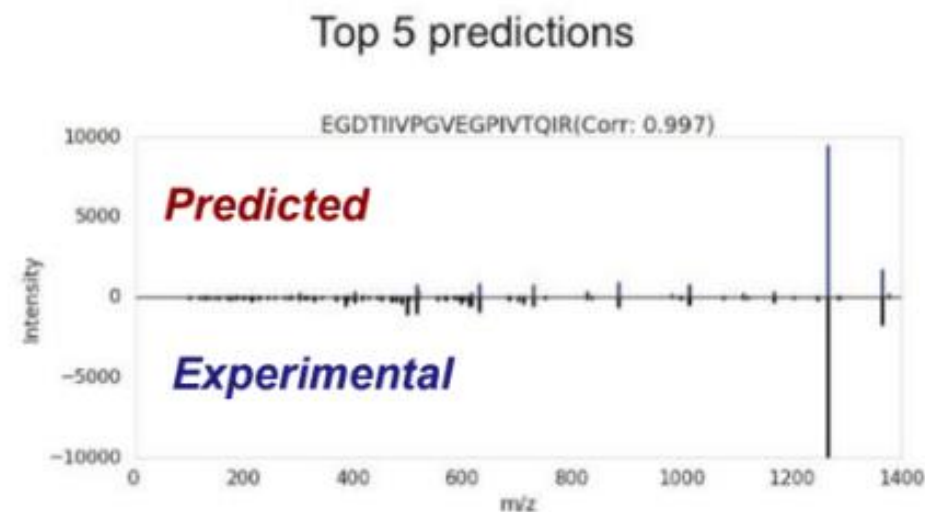
“ SAE (Sum of Absolute Errors)를 사용하되,
부족분의 잔차가 덜생기도록 가중치 부여 ”

Genome



Simulator

$$\begin{aligned}
 &\text{score (cost function)} \\
 &= \sum |y - \hat{y}| * 5 \text{ (if } y - \hat{y} > 0 \text{: under peak)} \\
 &\quad + \\
 &\quad \sum |y - \hat{y}| * 2 \text{ (if } y - \hat{y} < 0 \text{: over peak)}
 \end{aligned}$$





Thank You