리튬이온 배터리의
SOH(State Of Health, 수명상태) 추정방법

Presented By
양지안

목차

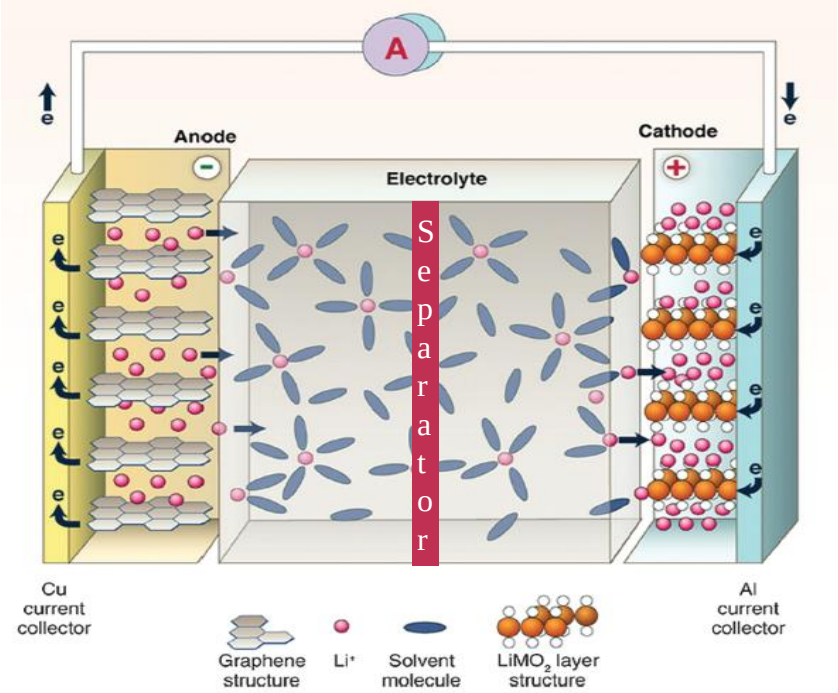
- 리튬이온 배터리의 동작 메커니즘 설명
- 배터리 상태 종류 및 상태 추정의 필요성
- 데이터 기반 이전의 상태 추정 방법론
- 데이터 기반의 상태 추정 방법론 (논문)
- 소감 및 추후 연구 내용

CVSP

리튬이온 배터리의 동작 메커니즘 설명

- **배터리 내부 구조 및 역할**

<배터리 내부 구조 및 방전 시 이온과 전자의 흐름>

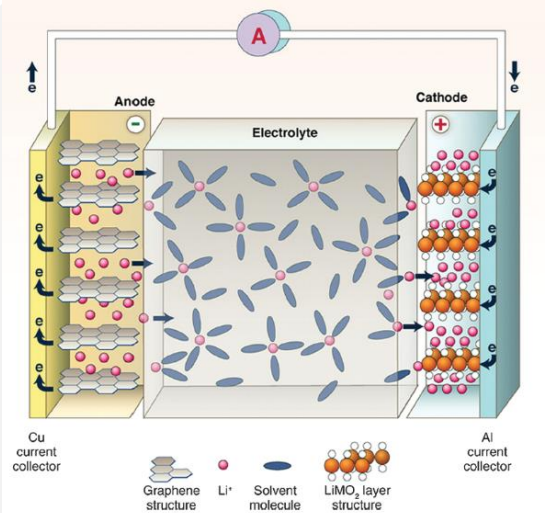


리튬이온배터리 내부의 집전체 (출처 : Mater. Express, Vol. 2, No. 3, 2012, 197-212)

	음극 (Anode)	전해질 (Electrolyte)	분리막 (Separator)	양극 (Cathode)	집전체
소재	흑연 등	염, 용매, 첨가제	고분자소재	NCM, LFP 등	구리(Anode), 알루미늄(Cathode)
역할	방전 시 리튬이온 방출 (충전 시 : 반대)	리튬이온 이동 매개체	쇼트 방지 (이온은 통과 가능, 전자는 통과 불가)	방전 시 리튬이온 흡수 (충전 시 : 반대)	전자의 이동통로

리튬이온 배터리의 동작 메커니즘 설명

- **배터리 동작 원리와 전압**



리튬이온배터리 내부의 집전체 (출처 : Mater. Express, Vol. 2, No. 3, 2012, 197-212)

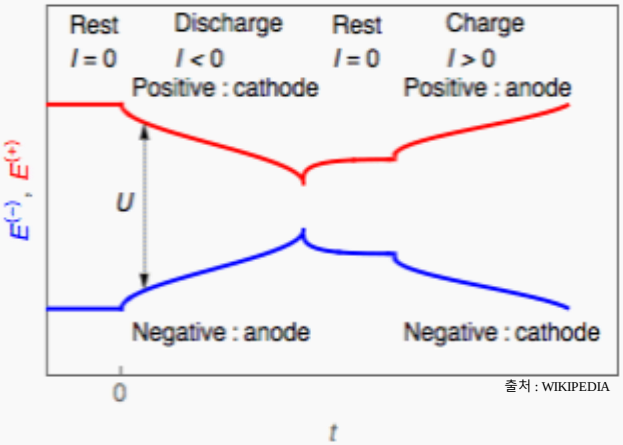
- 리튬 표준 전위 : - 3.04 V
  (전위에서의 - : 강한 환원제(전자를 버리고 싶어함)를 의미)

- 리튬(양극)보다 흑연(음극)의 전위가 더 높기 때문에
  방전(자연적인 화학반응) 방향은 음극 (Anode) -> 양극 (Cathode)가 됨

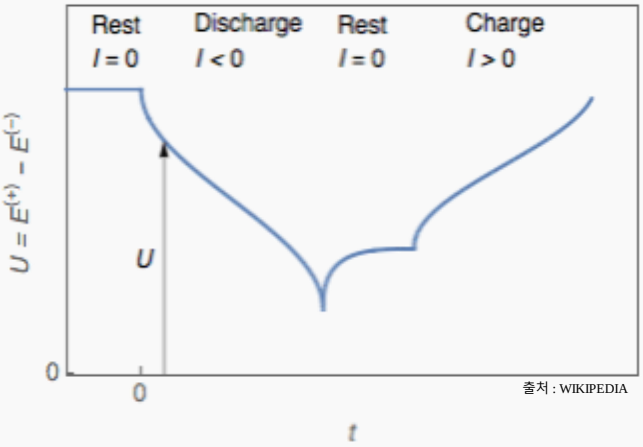
- 이 때 흑연과 리튬의 전위차 : 전압
  >> 배터리를 사용할 수 있는 범위를 표현

전기 화학적 반응

전기 화학적 반응을
전압으로 표현



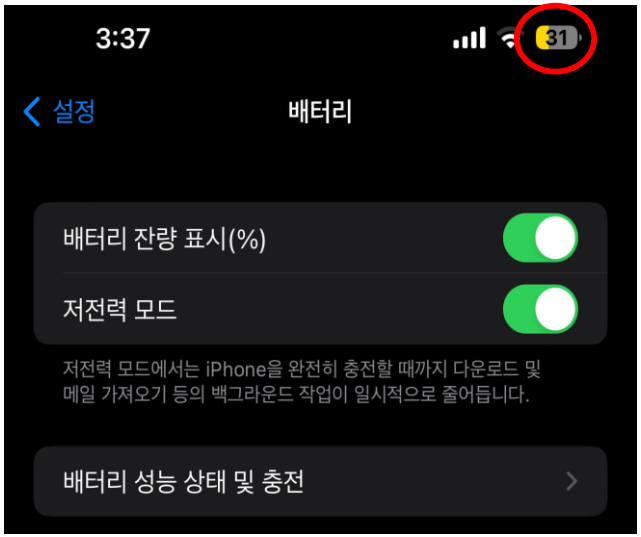
<충·방전 시 양극과 음극의 전위 변화>



<충 · 방전 시 배터리의 전압 변화>

배터리 상태 종류 및 상태 추정의 필요성

- **배터리 충전 상태 (SOC, State Of Charge)**



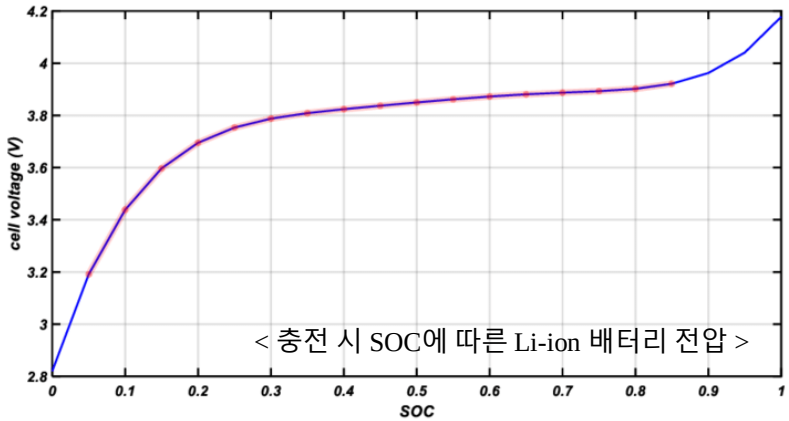
< 아이폰에서 충전상태 표시 >

✓ 충 · 방전에 따라 전압은 변화하지만, **전압의 변화 ≠ 충 · 방전한 양**

✓ **충 · 방전한 양** : 공칭 용량 (배터리 내 이동가능한 리튬이온의 총량) 대비 얼마나 사용했는가 라는 의미

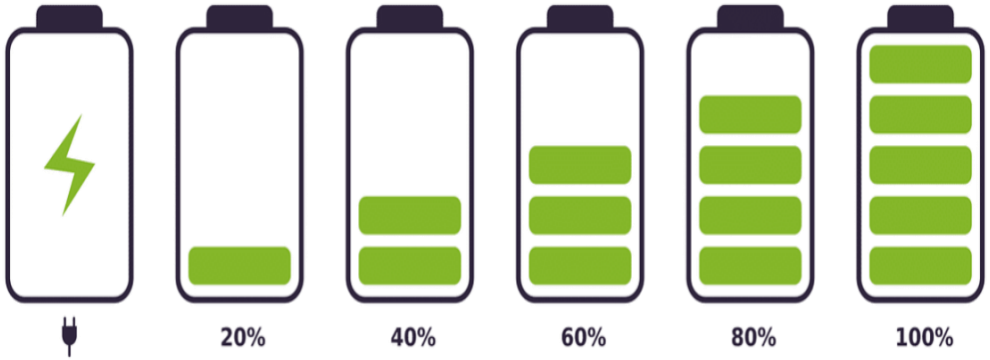
' **충전 상태 (SOC)** ' 로 표현함

$$\text{SOC}\,[\%] = \frac{\text{사용한 용량}\,[\text{Ah}]}{\text{공칭 용량}\,[\text{Ah}]} \times 100$$



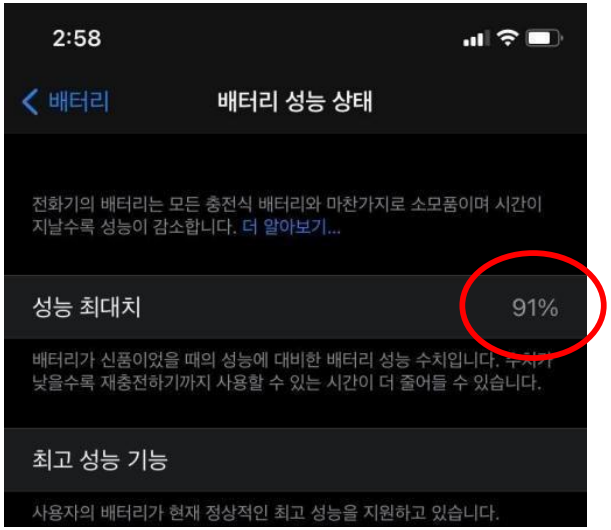
< 충전 시 SOC에 따른 Li-ion 배터리 전압 >

출처 : Battery Management System Performance Enhancement using Single Sliding Mode Based Charge Equalization Controller, *IOPSCIENCE*

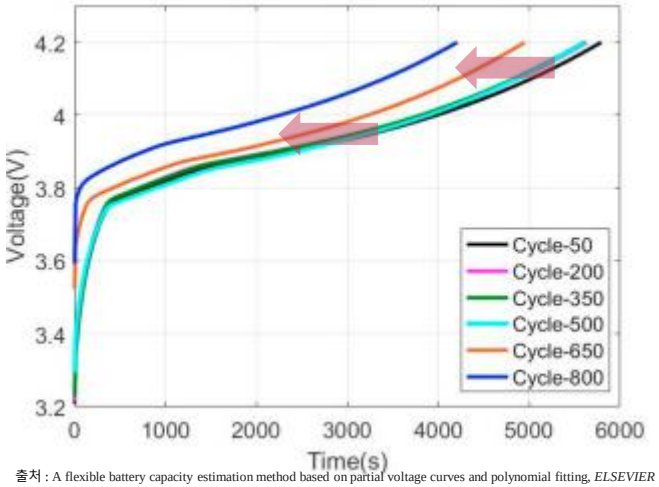


배터리 상태 종류 및 상태 추정의 필요성

- **배터리 수명 상태 (SOH, State Of Health)**



< 아이폰에서 수명상태 표시 >



출처 : A flexible battery capacity estimation method based on partial voltage curves and polynomial fitting, *ELSEVIER*

< 충 · 방전 반복 횟수에 따른 사용 시간 변화>

✓ 배터리는 충 · 방전을 반복할수록 내부 저항이 증가하여 **가용용량**이 줄어듦

✓ 이때, 공칭 용량 대비 **가용 용량**의 정도 : '**수명상태, 건강상태 (SOH)** '

$$\text{SOH}\,[\%] = \frac{\text{가용 용량}\,[\text{Ah}]}{\text{공칭 용량}\,[\text{Ah}]} \times 100$$

$$\text{SOC}\,[\%] = \frac{\text{사용한 용량}\,[\text{Ah}]}{\text{가용 용량}\,[\text{Ah}]} \times 100$$



SOH : 40 %
SOC : 100 %

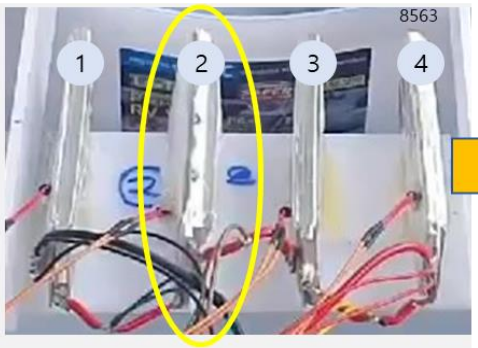
SOH : 60 %
SOC : 100 %

SOH : 80 %
SOC : 100 %

SOH : 100 %
SOC : 100 %

배터리 상태 종류 및 상태 추정의 필요성

- **배터리 상태 추정의 필요성**



20 s



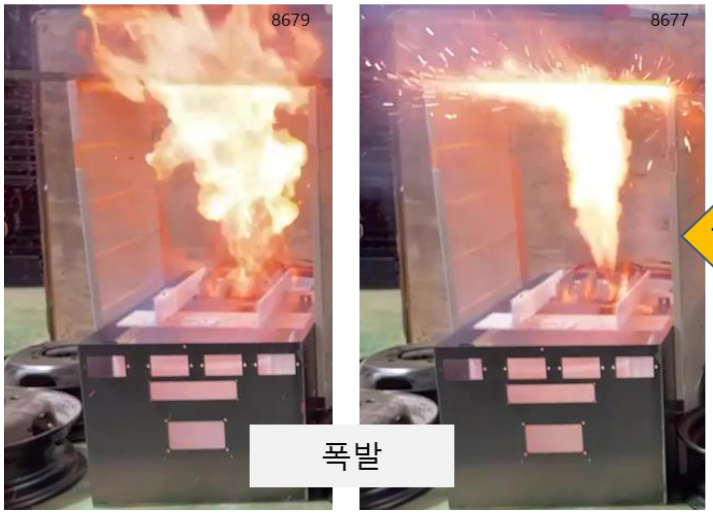
35 s



1 s



1 s 이내



- SOH (수명상태)를 잘못 추정할 경우 >> 가용용량 추정 오류 >> SOC (충전상태) 추정 오류 발생 >>
  >> 과충전 및 과방전 사용 위험 증가 >> 화재 발생 위험 증가

CVSP

데이터 기반 이전의 상태 추정 방법론

- **배터리 충전 상태 (SOC, State Of Charge)**

1. 전류적산법

$$\mathrm{SOC(t)} = \mathrm{SOC}(t_0) + \int_0^t \frac{\mathrm{I}}{\mathrm{C_{bat}}} \cdot \mathrm{dt}$$

Where:

$SOC(t)$: Battery State Of Charge at time t [%]

$\mathrm{SOC}(t_0)$: Battery initial State Of Charge [%]

I : Charge / Discharge current [A]

t : Time [h]

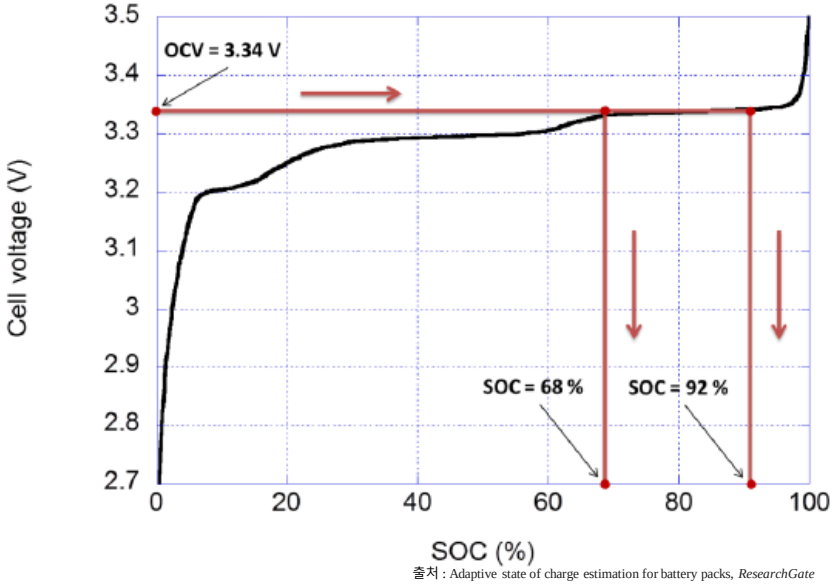
$\mathrm{C_{bat}}$: Battery Capacity [Ah]

- 단점

1. 전류를 측정할 때 발생하는 센서의 오차가 지속 누적

2. initial SOC를 알아야 함
= initial SOC가 완전 충전과 완전 방전 전압이어야 오차 감소

2. SOC-OCV Table Matching



출처 : Adaptive state of charge estimation for battery packs, *ResearchGate*

< SOC 별 개방회로 전압(OCV) 곡선>

- SOC-OCV Table matching
: 개방회로 전압 (OCV, Open Circuit Voltage) 을 SOC matching하여 SOC 추정

- 단점

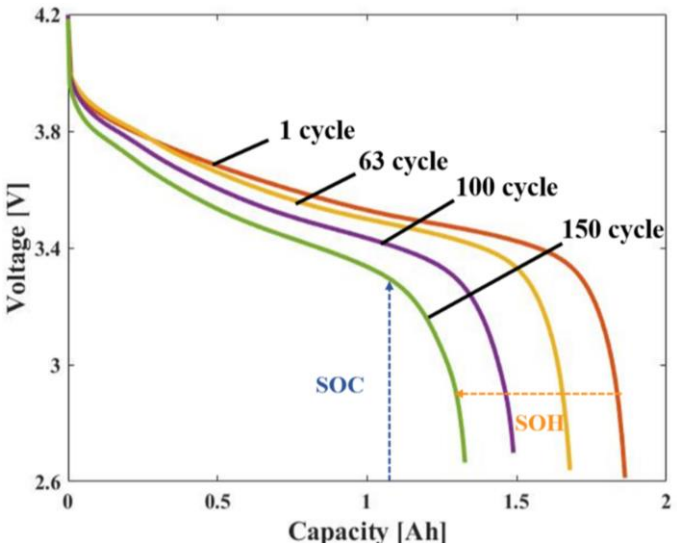
1. OCV 상태까지 시간이 오래 걸림

2. 전압이 Flat한 배터리는 추정이 힘듦
= LFP의 SOC 중간영역에서 전압이 Flat한 구간 존재함

데이터 기반 이전의 상태 추정 방법론

- **배터리 수명 상태 (SOH, State Of Health)**

1. 용량 측정법



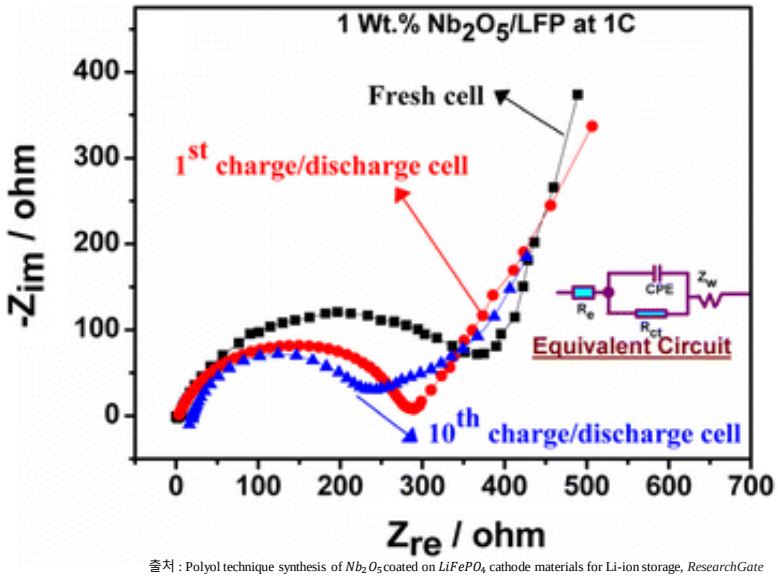
출처 : Estimation of Online State of Charge and State of Health Based on Neural Network Model Banks Using Lithium Batteries, *MDPI*

< Cycle 횟수 별 SOH 변화>

- 용량 측정법
  : 직접 충방전을 진행하여 용량을 측정하는 방법
- 단점

  1. 용량을 측정하면서 배터리에 영향을 주어 수명 단축
  2. 시간 소모 큼

2. 임피던스 측정법



출처 : Polyol technique synthesis of Nb_2O_5 coated on $LiFePO_4$ cathode materials for Li-ion storage, *ResearchGate*

< Cycle 횟수 별 내부 저항의 변화>

- 임피던스 측정법
  : 다양한 주파수를 인가하여 내부 저항을 측정하는 방법
- 단점

  1. 환경적 요인이나 측정 조건에 따라 값이 달라질 수 있음
  2. 배터리에 전류가 인가되지 않은 상태여야 함

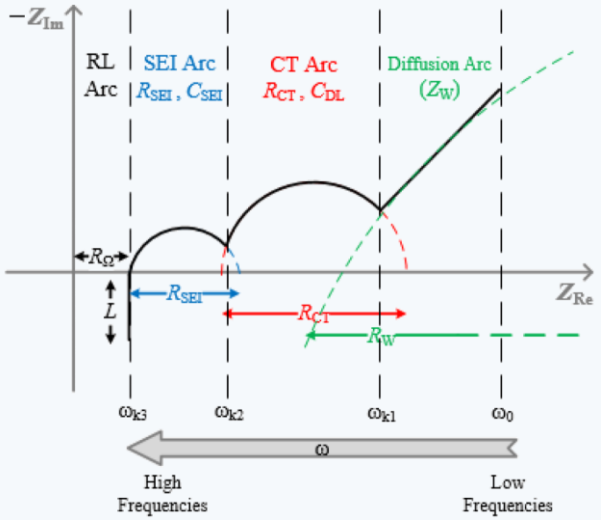
데이터 기반 이전의 상태 추정 방법론

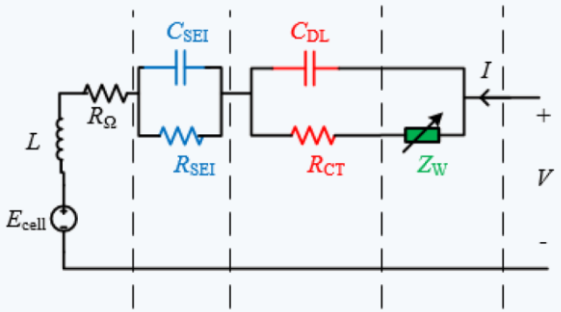
- 배터리 모델링

1. 등가회로 모델 (ECM, Equivalent Circuit Model)

1

< 임피던스 측정법으로 도출되는 배터리 내부 저항>



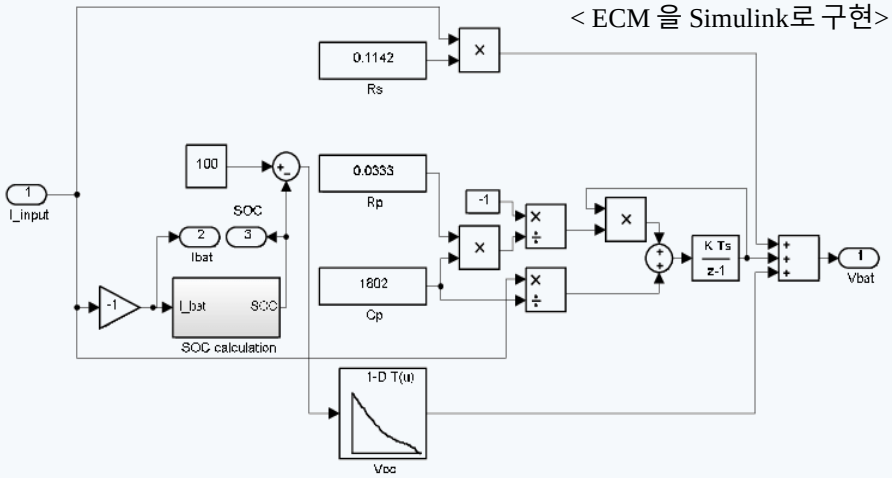


< 배터리를 전기적으로 표현한 등가회로모델 (ECM)>

출처 : Robust Approach to Battery Equivalent-Circuit-Model Parameter Extraction Using Electrochemical Impedance Spectroscopy, *MDPI*

2

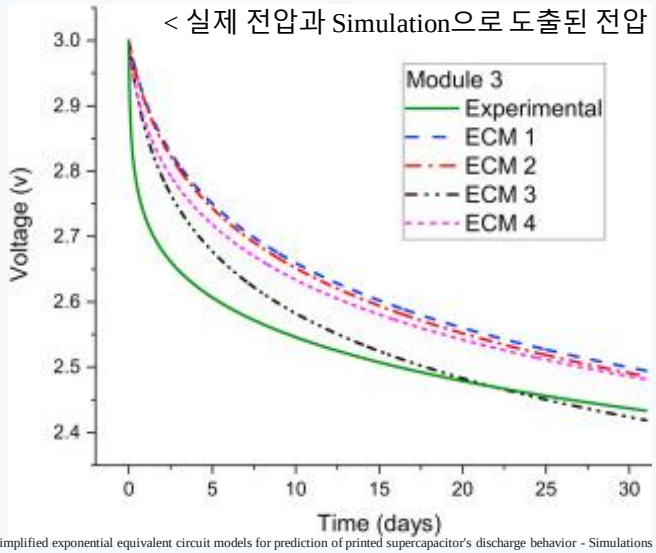
< ECM 을 Simulink로 구현>



출처 : Modeling and simulation of first-order Li-Ion battery cell with experimental validation, *Research Gate*

3

< 실제 전압과 Simulation으로 도출된 전압 >



출처 : Simplified exponential equivalent circuit models for prediction of printed supercapacitor's discharge behavior - Simulations and experiments, *ELSEVIER*

CVSP

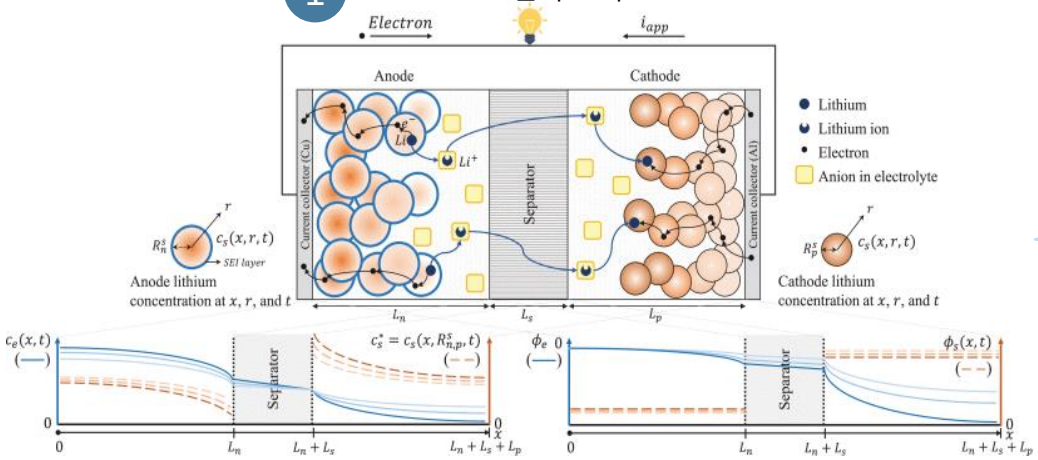
데이터 기반 이전의 상태 추정 방법론

- 배터리 모델링

2. Pseudo-Two-Dimensional 모델 (P2D Model)

1

< P2D 모델의 모식도>



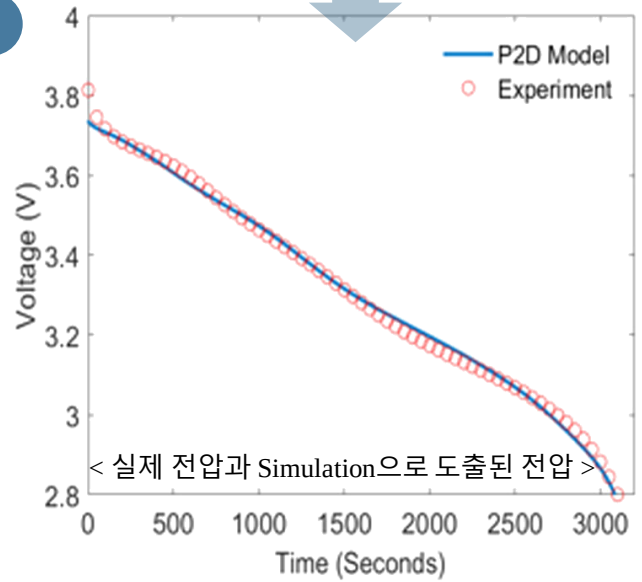
출처 : Effective and practical parameters of electrochemical Li-ion battery models for degradation diagnosis, *ELSEVIER*

Symbol	Parameter	Positive Electrode	Separator	Negative Electrode	Units
σ_i	Solid phase conductivity	100		100	S/m
$\varepsilon_{f,i}$	Filler fraction	0.025		0.0326	-
ε_i	Porosity	0.385	0.724	0.485	-
$Brugg$	Bruggman Coefficient	4	4	4	-
D	Electrolyte diffusivity	7.5×10^{-10}	7.5×10^{-10}	7.5×10^{-10}	m^2/s
D_i^s	Solid Phase Diffusivity	1.0×10^{-14}		3.9×10^{-14}	m^2/s
k_i	Reaction Rate constant	2.334×10^{-11}		5.031×10^{-11}	$m^{2.5}/(mol^{0.5}\,s)$
$c_{i,\max}^s$	Maximum solid phase concentration	51554		30555	mol/m^3
$c_{i,0}^s$	Initial solid phase concentration	25751		26128	mol/m^3
c_0	Initial electrolyte concentration	1000	1000	1000	mol/m^3
$R_{p,i}$	Particle Radius	2.0×10^{-6}		2.0×10^{-6}	m
a_i	Particle Surface Area to Volume	885000		723600	m^2/m^3
l_i	Region thickness	80×10^{-6}	25×10^{-6}	88×10^{-6}	m
t_+	Transference number		0.364		-
F	Faraday's Constant		96487		C/mol
R	Gas Constant		8.314		J/(mol·K)
T	Temperature		298.15		K
ρ	Density	2500	1100	2500	kg/m^3
k_{SEI}	Rate constant for SEI			1.5×10^{-6}	A/m^2
α	Transfer coefficient	0.5		0.5	-
U_{SEI}	Open circuit potential for SEI layer			0.4	V
κ_{SEI}	Conductivity of SEI layer			1	S/m
ρ_{SEI}	Density of SEI layer			2.1×10^3	kg/m^3
M_{SEI}	Molecular weight of SEI			0.026	kg/mol

출처 : Integrating Energy Storage Systems into Renewable Grid Applications: A Model-based Approach, *Research Gate*

< P2D 모델 구현에 필요한 파라미터>

2



< 실제 전압과 Simulation으로 도출된 전압 >

출처 : A Robust and Sleek Electrochemical Battery Model Implementation: A MATLAB® Framework, *ECS*

CVSP

데이터 기반의 상태 추정 방법론

- 참고 논문 **A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium- Ion Battery Remaining Useful Life**

Lei Ren , Member, IEEE, Jiabao Dong , Xiaokang Wang , Member, IEEE, Zihao Meng, Li Zhao, and M. Jamal Deen , Fellow, IEEE
IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 17, NO. 5, MAY 2021

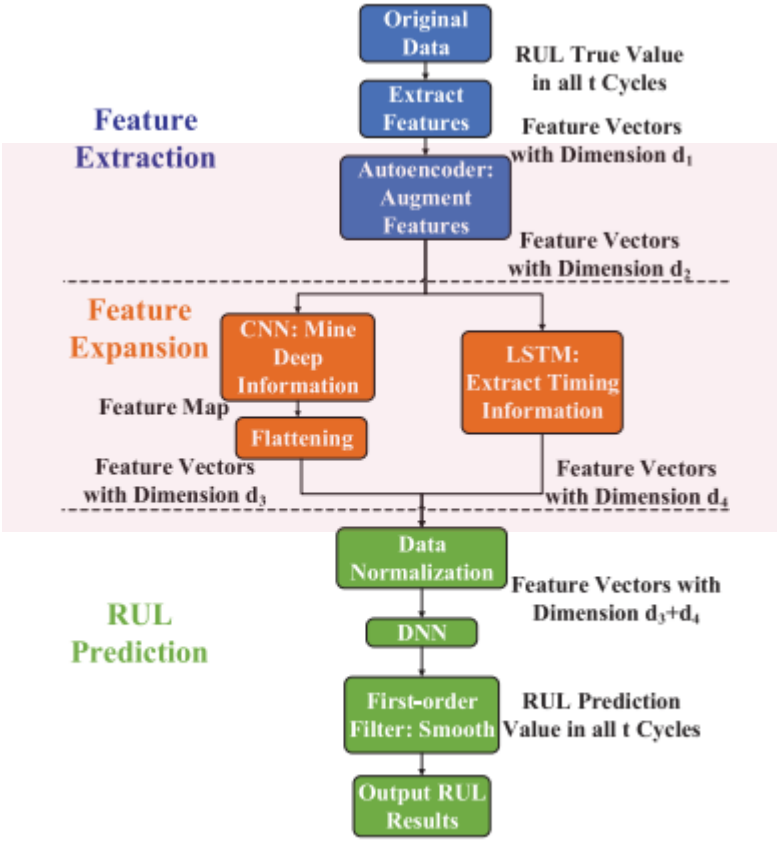


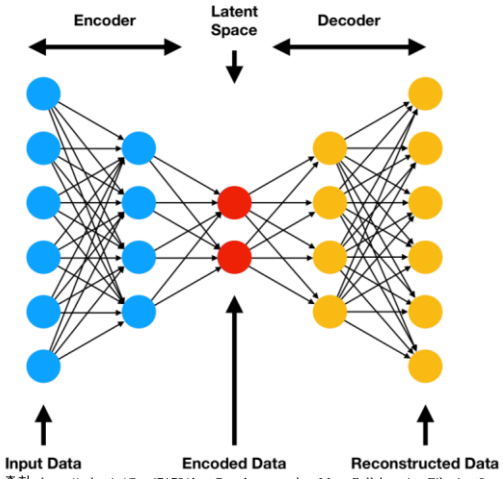
Fig. 1. Framework for LIB RUL prediction.

✓ 논문에서 Main이 되는 방법론

1. Autoencoder
2. CNN
3. LSTM

데이터 기반의 상태 추정 방법론

- **Autoencoder**



출처 : https://velog.io/@smj7172/AutoRec-Autoencoders-Meet-Collaborative-Filtering-0

✓ **기존 Autoencoder :**
입력 데이터를 압축 및 축소 → latent vector 추출 → 원본 데이터와 유사하게 복구

- 동작 단계

1. 인코딩 : 입력데이터 $\boldsymbol{x}$가 encoder → latent vector $\boldsymbol{z}$로 변환, 데이터의 차원이 축소

2. 디코딩 : Latent vector $\boldsymbol{z}$가 decoder → 원본 데이터와 유사한 재구성 데이터 $\hat{\boldsymbol{x}}$로 변환, 데이터의 차원이 다시 확장

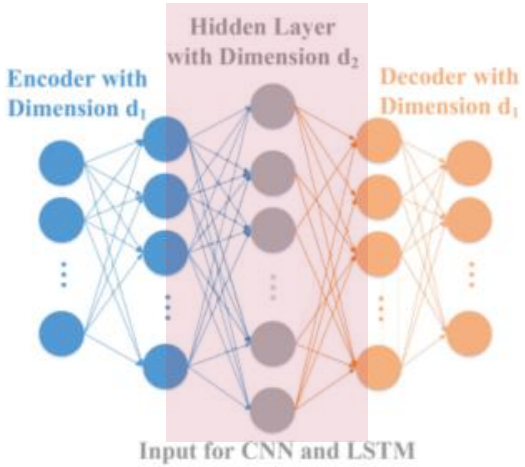


Fig. 2. Autoencoder structure in Auto-CNN-LSTM.

✓ **논문에서의 Autoencoder** : **특징 벡터의 차원 증가를 위해 사용**

>> 원본 데이터가 풍부하지 않거나, 추가 정보 추출이 필요할 때

>> '차원의 저주' 와 같은 문제 가능성 발생

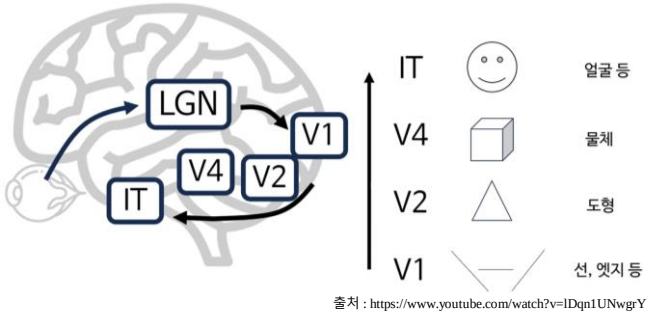
>> Autoencoder에서 추출된 데이터를 CNN과 LSTM에 적용하면서 문제점 완화

CVSP

데이터 기반의 상태 추정 방법론

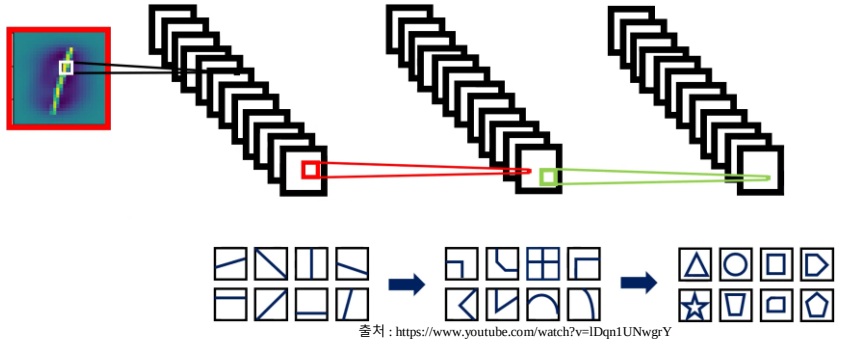
- **합성곱신경망 (CNN, Convolutional Neural Network) – 전체**

  : 인간의 시신경을 모방하여 만든 딥러닝 구조 중 하나로 공간적인 정보를 유지하면서 특성을 추출하는 방법



출처 : https://www.youtube.com/watch?v=lDqn1UNwgrY

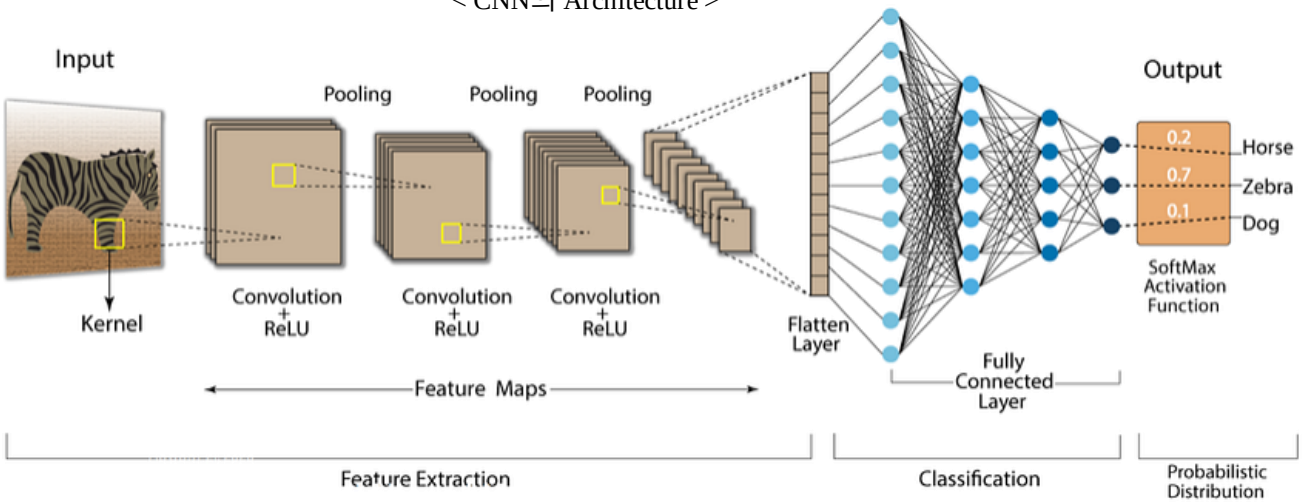
< 인간의 시각정보 처리과정 >



출처 : https://www.youtube.com/watch?v=lDqn1UNwgrY

< 합성곱 층의 깊이에 따른 복잡한 feature 처리 과정 >

< CNN의 Architecture >

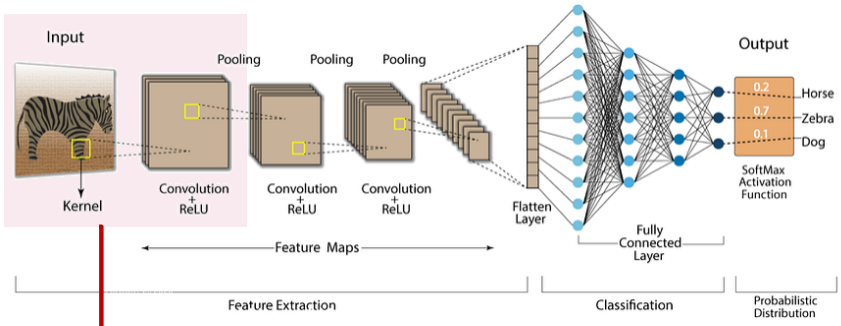


출처 : https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5

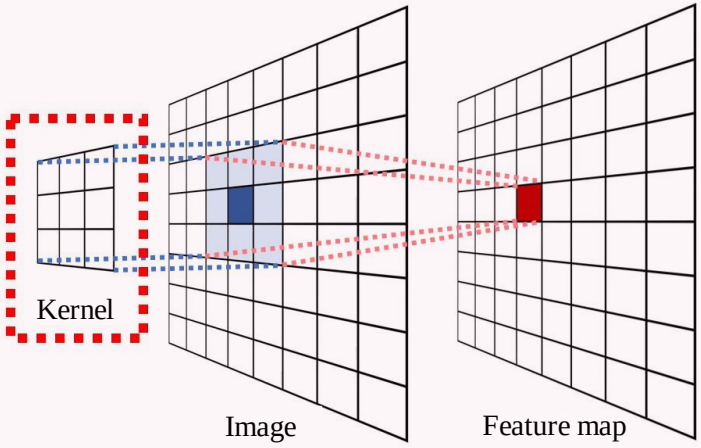
데이터 기반의 상태 추정 방법론

- **합성곱신경망 (CNN, Convolutional Neural Network) – Convolution layer**

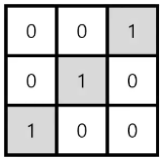
  : 원본 이미지의 특성을 나타내는 Feature map을 만드는 단계



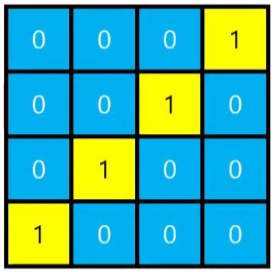
작은 크기의 Kernel (필터)로 합성곱 연산을 통해 Feature map 생성



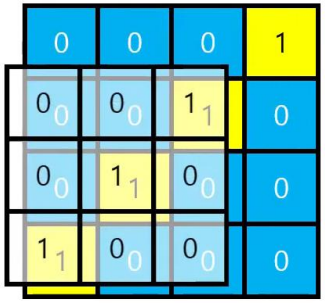
< CNN의 Convolution layer >



Kernel 1

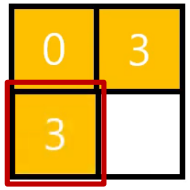


Image



Image

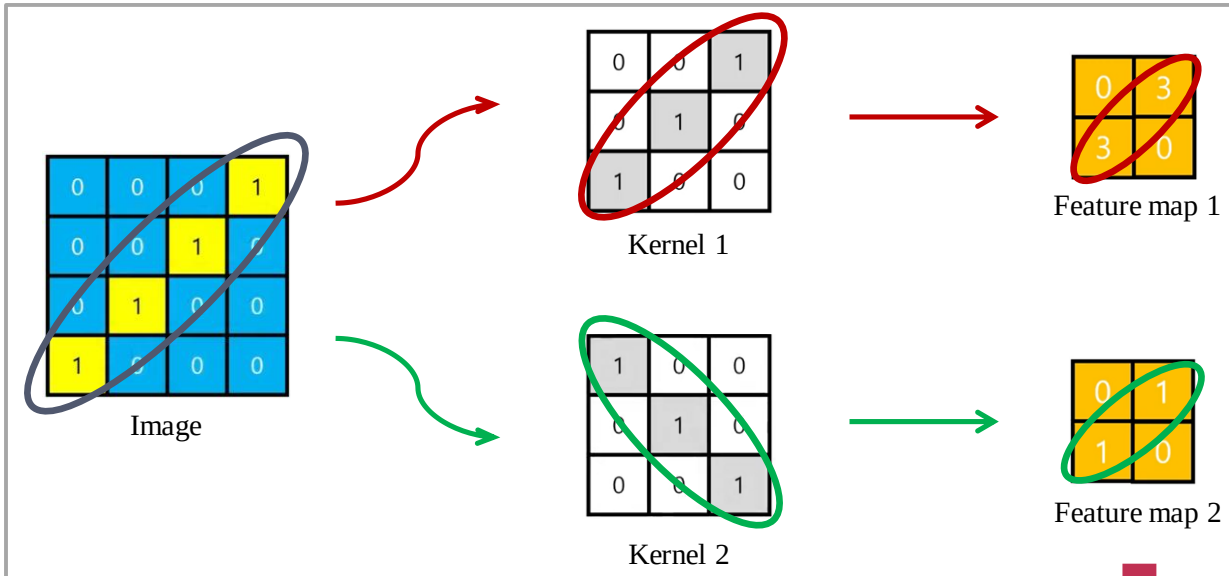
$(0 \times 0) + (0 \times 0) + (1 \times 1) +$
$(0 \times 0) + (1 \times 1) + (0 \times 0) +$
$(1 \times 1) + (0 \times 0) + (0 \times 0) = 3$



Feature map 1

데이터 기반의 상태 추정 방법론

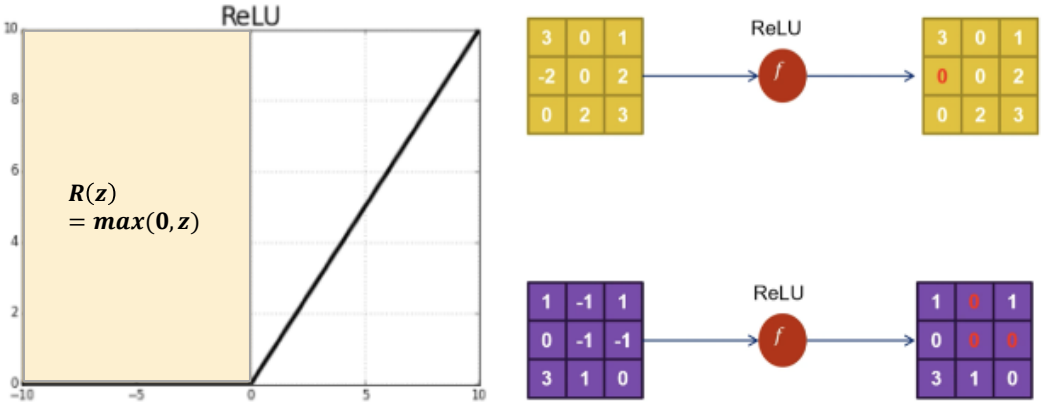
- **합성곱신경망 (CNN, Convolutional Neural Network) – Convolution layer**



▫ Kernel 과 Image가 비슷한 형태이면 Feature map 의 값이 높게 나타남

✓ ReLu (Rectified Linear Unit) 함수
z > 0 : z 를 출력
z ≤ 0 : 0 을 출력

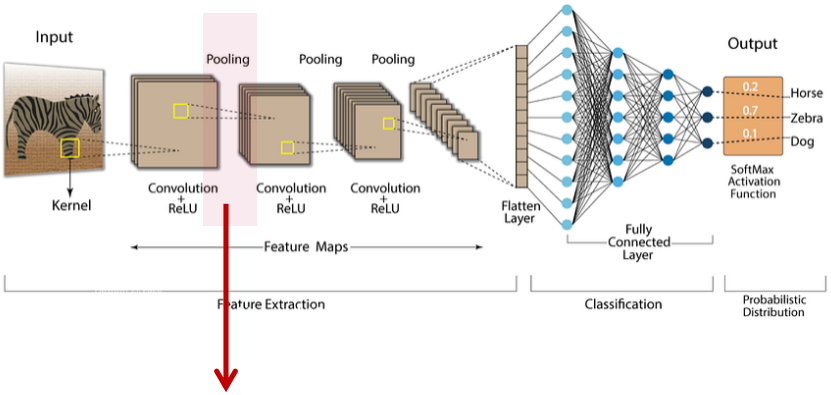
✓ Feature map의 값이 낮은 부분은 불필요한 정보이므로 ReLu 함수를 이용해 0으로 변환



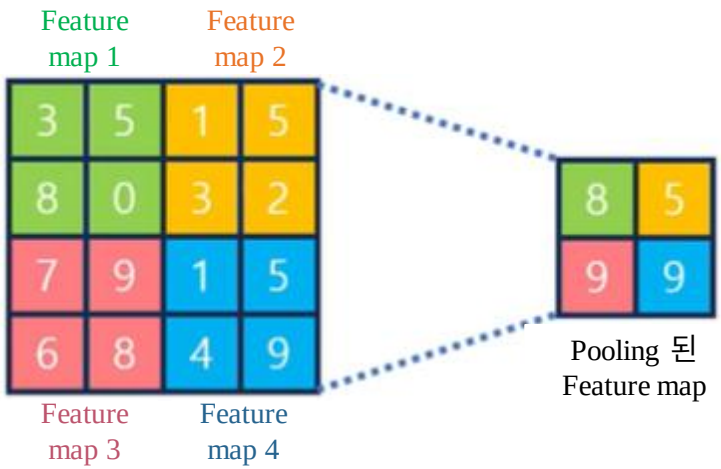
데이터 기반의 상태 추정 방법론

- **합성곱신경망 (CNN, Convolutional Neural Network) – Pooling layer**

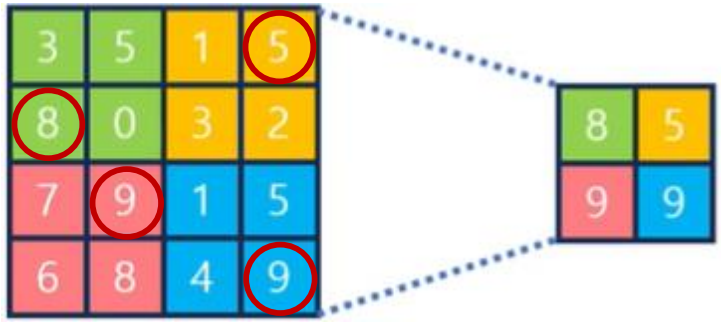
  : Convolution layer에서 추출된 Feature map의 크기(공간적인 크기)를 줄이는 단계



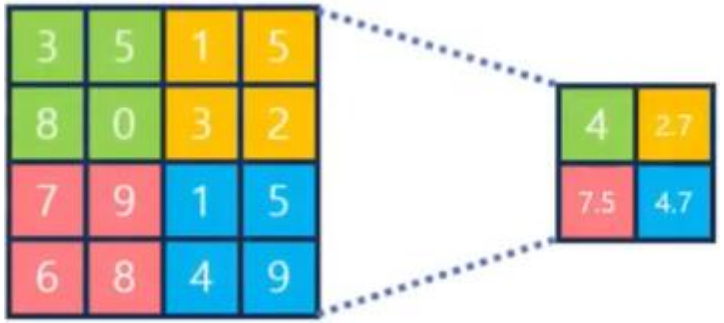
Pooling layer



Pooling 된 Feature map



방법 1. Max Pooling layer

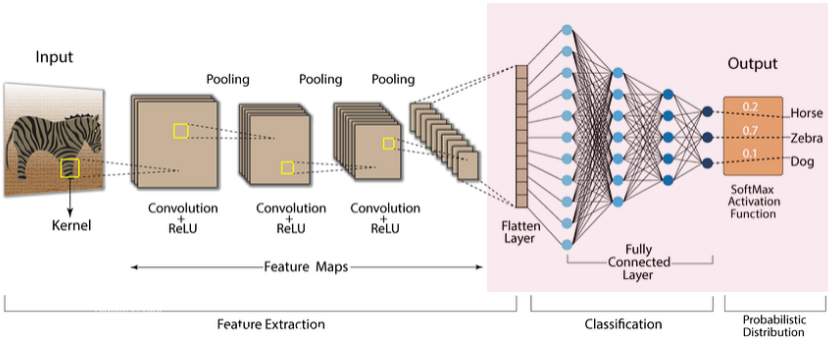


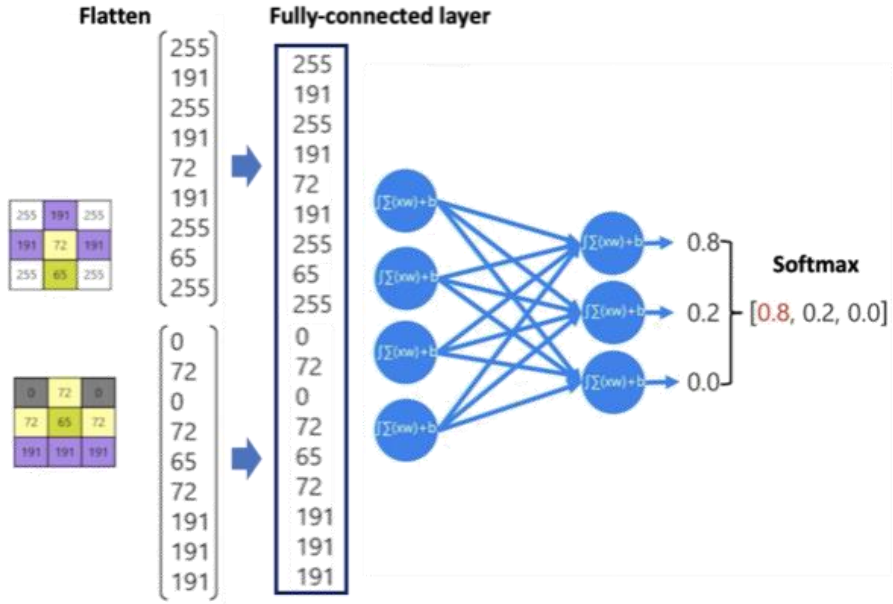
방법 2. Average Pooling layer

데이터 기반의 상태 추정 방법론

- **합성곱신경망 (CNN, Convolutional Neural Network) – Flatten layer & Fully connected layer & Output**

  : Feature map을 정리하여 분류 및 출력하는 단계





✓ Flatten layer (평탄화 계층)

: **2차원**의 Feature map을 **1차원**의 벡터로 변형하여 분류에 적합한 형태로 **변환**

✓ Fully connected layer

: 데이터의 분석 혹은 분류를 위해 가중치를 곱하고 편향을 더하여 출력을 계산

✓ Softmax 함수

: Fully connected layer로부터 출력된 값을 확률 분포로 변환하여 변환된 값 중 높은 값을 통해 이미지를 분류

데이터 기반의 상태 추정 방법론

- **장단기 메모리 (LSTM, Long Short-Term Memory) – 발생 배경**

: 순환신경망(RNN, Recurrent Neural Network)의 한 종류로 장기 기억과 단기 기억을 조합하여 시계열 데이터를 처리하는 방법

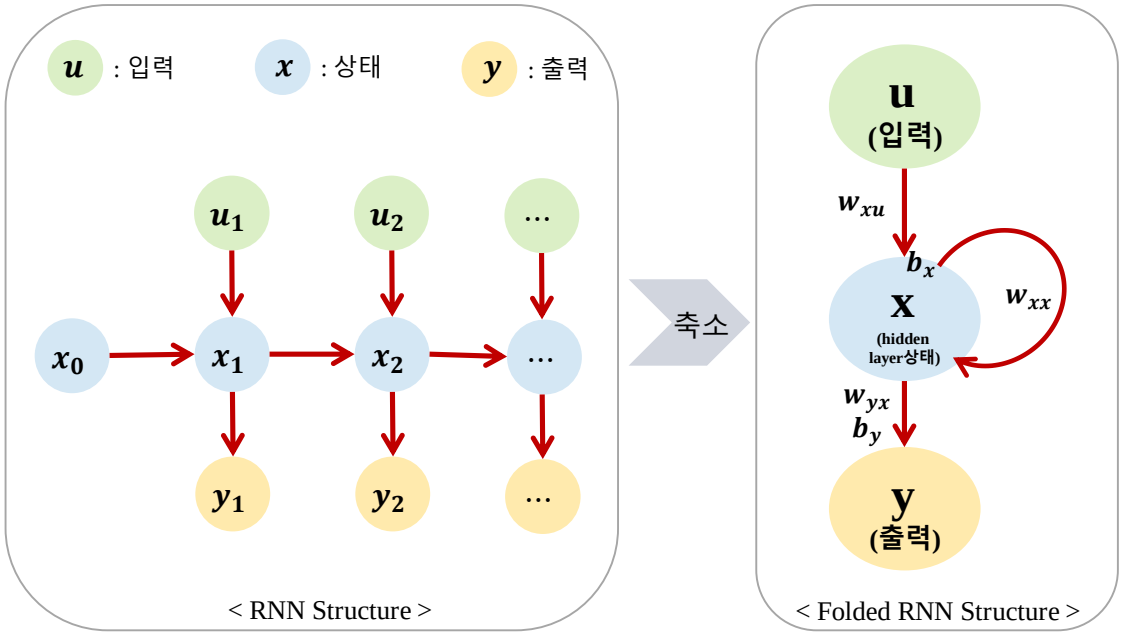
✓ 순환신경망 (RNN)

: 과거의 정보와 새롭게 입력되는 정보를 연결하여 시간의 흐름에 따라 변화하는 데이터를 학습하는 방법

◦ 현재 시간의 상태 $(\boldsymbol{x_t})$ 가 이전 시간의 상태 $(\boldsymbol{x_{t-1}})$ 와 현재의 입력 $(\boldsymbol{u_t})$에 관계가 있는 경우 :

$$\boldsymbol{x_t = f\,(x_{t-1}, u_t)} \;\rightarrow\; \boldsymbol{x_t = \sigma\,(W_{xx}x_{t-1} + W_{xu}u_t + b_x)}$$

◦ 각 시간에서 관측 가능한 상태의 모음은 출력 $(\boldsymbol{y_t})$: $\boldsymbol{y_t = h\,(x_t)} \rightarrow \boldsymbol{y_t = \sigma\,(W_{yx}x_t + b_y)}$



✓ 순환신경망 (RNN) 한계점

: RNN 구조에서 State $\boldsymbol{x_t}$에 $\boldsymbol{w_{xx}}$ 가 계속 곱해짐

$$w_{xx}{}^{t} x_0 = x_t$$

① 곱해지는 $\boldsymbol{w_{xx}}$ 값 > 1 : ∞ → 기울기 폭주

② 곱해지는 $\boldsymbol{w_{xx}}$ 값 < 1 : 0 → 기울기 소실

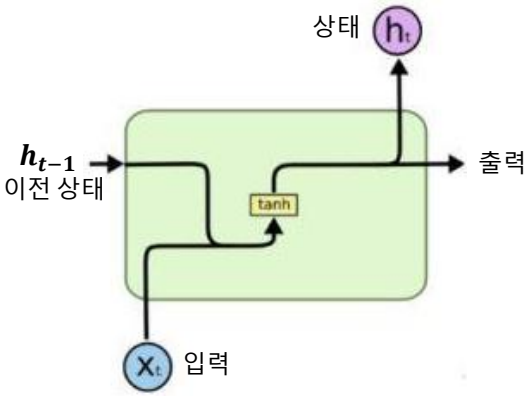
✓ 해결방안

① 기울기 폭주 : Gradient clipping
(Gradient가 일정 threshold를 넘어가면 기울기 값을 자르는 방법)

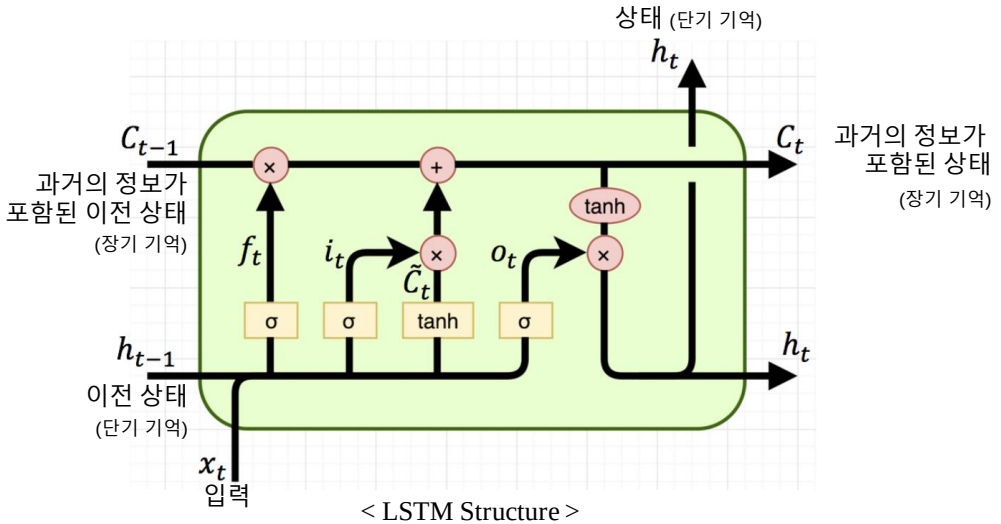
② 기울기 소실 : Gate를 추가한 RNN
→ LSTM

데이터 기반의 상태 추정 방법론

- **장단기 메모리 (LSTM, Long Short-Term Memory) – 전체**
  : 중요한 정보 기억을 제어할 수 있는 Gate가 추가된 RNN



< RNN Structure >



< LSTM Structure >

✓ LSTM의 핵심 아이디어

▫ Cell State (C_t) : 컨베이어 벨트와 같이 흘러가면서 정보를 효율적으로 전달 및 저장하는 부분 추가

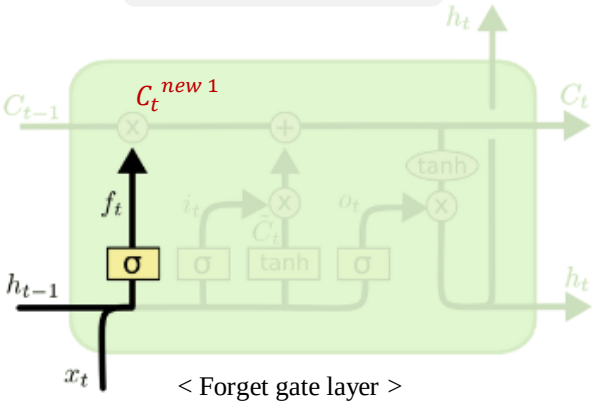
>> 정보가 전달되는 과정에서 **다양한 Gate (Input gate , Forget gate, Output gate)**를 통해
중요한 정보는 장기간 보존하고 불필요한 정보는 버리면서 **필요한 정보를 선택적으로 처리 및 기억**

데이터 기반의 상태 추정 방법론

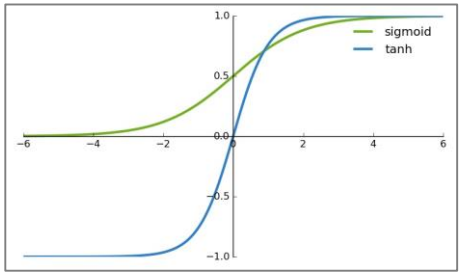
- **장단기 메모리 (LSTM, Long Short-Term Memory) – Forget gate & Input gate**

  : 잊어버릴 정보를 처리하는 Forget gate & 업데이트 할 정보를 처리하는 Input gate

1. Forget gate



< Forget gate layer >



< Sigmoid & Tanh 함수>

- Sigmoid : 강조를 조절하는 밸브 역할
  (1: 열림, 0: 닫힘)
- tanh : Cell state에 추가될 정보의 내용 결정

① $f_t = \sigma\,(W_f \cdot [h_{t-1}, x_t] + b_f)$

(기존의 정보에서 어떤 정보를 얼마나 버릴지 계산)

- f_t = 0 : C_{t-1} 모두 버림
- f_t = 1 : C_{t-1} 모두 유지

② $C_t^{new\ 1} = f_t \odot C_{t-1}$

(C_{t-1} (기존 정보) 요소를 개별적으로 버릴지 말지 계산)

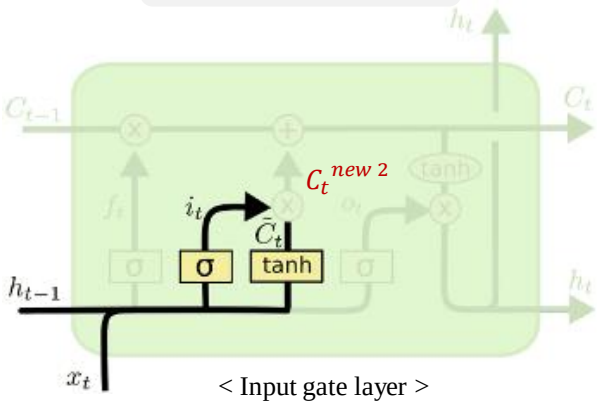
Ex)

f_t = [1,0.5,0] , C_{t-1} = [0.5,−0.1,0.8]

$f_t \odot C_{t-1}$ = [(1 · 0.5) , (0.5 · −0.1), (0 · 0.8)]

= [0.5, −0.05 ,0]

2. Input gate



< Input gate layer >

① $i_t = \sigma\,(W_i \cdot [h_{t-1}, x_t] + b_i)$

(얼마나 많은 정보를 추가할지 계산)

② $\tilde{C}_t = \tanh\,(W_c \cdot [h_{t-1}, x_t] + b_c)$

(어떤 새로운 정보를 업데이트 할지 계산,
Cell state 추가할 내용 후보군)

③ $C_t^{new\ 2} = i_t \odot \tilde{C}_t$

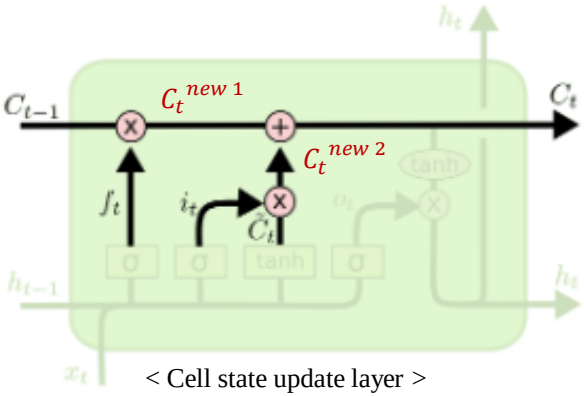
($\tilde{C}_t$ (업데이트 할 내용) 요소를 개별적으로
기존 Cell state에 얼마나 많이 반영할 지 계산)

데이터 기반의 상태 추정 방법론

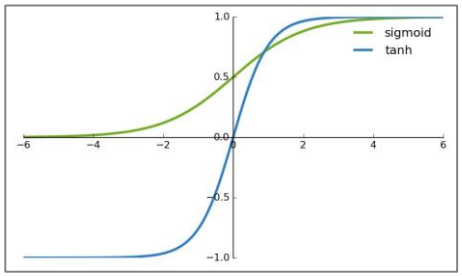
- **장단기 메모리 (LSTM, Long Short-Term Memory) – Cell state update & Output gate**

  : 필요한 정보를 남기는 Cell state update & 적절한 정보를 출력하는 Output gate

3. Cell state 업데이트

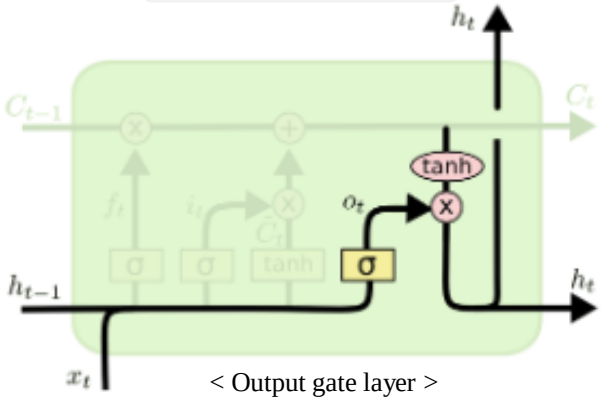


< Cell state update layer >



< Sigmoid & Tanh 함수>

4. Output gate



< Output gate layer >

- Sigmoid : 강조를 조절하는 밸브 역할
  (1: 열림, 0: 닫힘)
- tanh : Cell state에 추가될 정보의 내용 결정

① $C_t = (f_t \odot C_{t-1}) + (i_t \odot \tilde{C}_t)$

(필요한 정보를 보존하면서 새로운 정보를 받아들이는 과정)

- $C_t^{new\ 1} = f_t \odot C_{t-1}$
- $C_t^{new\ 2} = i_t \odot \tilde{C}_t$

→ $C_t = C_t^{new\ 1} + C_t^{new\ 2}$

① $o_t = \sigma\,(W_o \cdot [h_{t-1}, x_t] + b_o)$

(얼마나 많은 정보가 출력될지 계산)

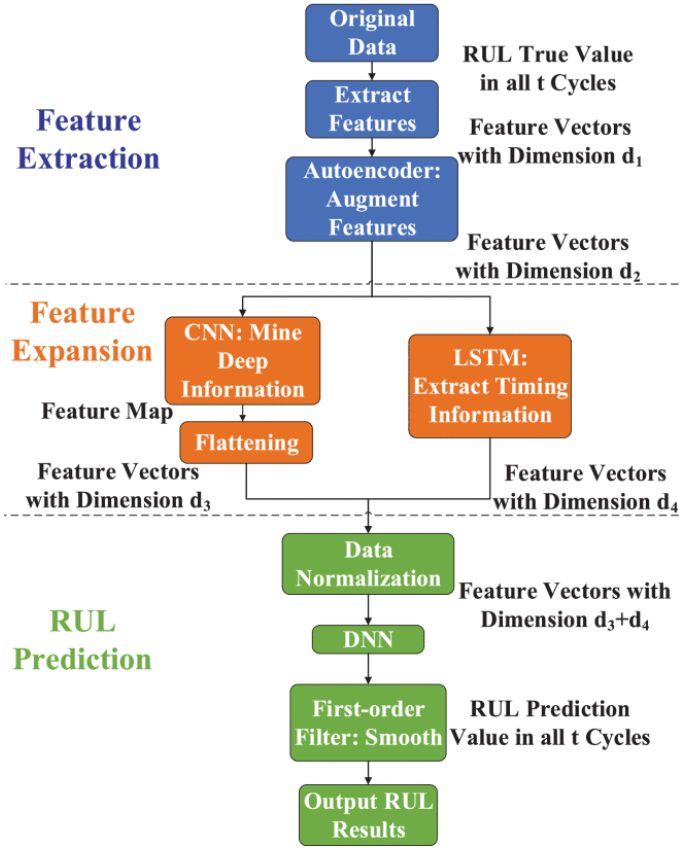
② $h_t = o_t \odot \tanh(C_t)$

(실제로 어떤 정보가 출력할 지 계산하는 과정으로 o_t가 얼마나 열렸는지에 따라 Cell state의 정보를 조절)

데이터 기반의 상태 추정 방법론

- **참고 논문** **A Data-Driven Auto-CNN-LSTM Prediction Model for Lithium- Ion Battery Remaining Useful Life**

Lei Ren , Member, IEEE, Jiabao Dong , Xiaokang Wang , Member, IEEE, Zihao Meng, Li Zhao, and M. Jamal Deen , Fellow, IEEE
IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 17, NO. 5, MAY 2021



1. Feature Extraction

1-1. Original Data : t번의 충·방전 Cycle 동안
RUL(남은 수명 상태 값)을 포함한 충·방전 데이터

1-2. Extract Features: 충·방전 데이터에서 추출한 특징 벡터 (d_1 size : 14 x 21)

1-3. Autoencoder : 특징 벡터 증가 (d_2 size : 14 x 50)

2. Feature Expansion

2-1-1. CNN : d_2 에서 더 복잡한 정보 추출

2-1-2. Flattening : CNN으로 추출된 값을 1차원 벡터로 평탄화하여 d_3 생성
(d_3 size : 1 x 480)

2-2. LSTM : d_2 에서 시간에 따른 특성과 변화를 학습하여 d_4 생성

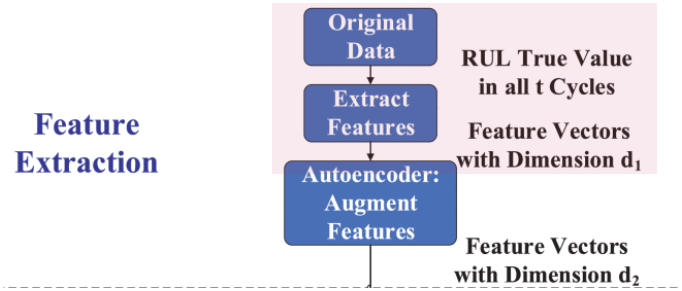
3. Feature Expansion

3-1. Data Normalization : 다양한 단위의 변수를 정규화하여 가중치 조정

3-2. DNN + Smoothing : 정규화된 Feature 값으로 RUL 예측

데이터 기반의 상태 추정 방법론

- **1-1 & 1-2 . Feature Extraction with Original data**



< Framework에서 Feature Extraction >

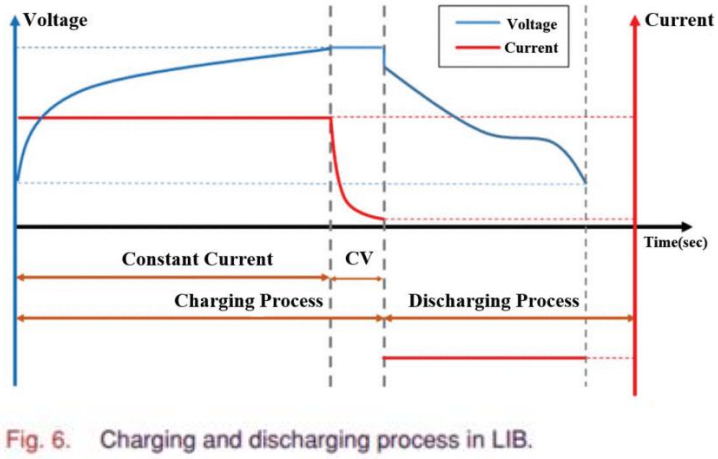


Fig. 6. Charging and discharging process in LIB.

✓ 배터리 SOH 추정에 사용된 데이터 설명

- 충전 : 4.2 V & 1.5 A (CC) & 20 mA (CV)

- 방전 : 2 A

- SOH가 70%로 떨어질 때까지 반복

- 사용된 데이터 : #5 ~ #7 (168 Cycle) & #25 ~ #28 (28 Cycle)

(각 Cycle 중 **14번의 연속된 충 · 방전Cycle**만 사용)

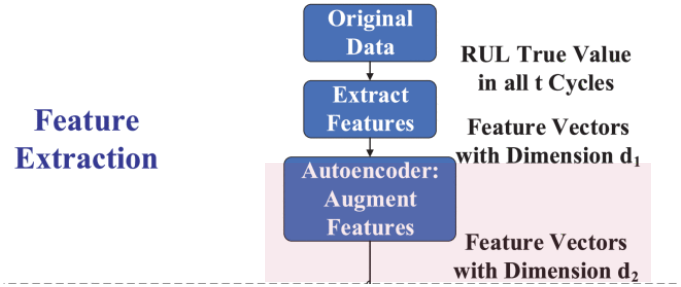
- Train : #5, #6 & #25, #26, #27
- Test : #7 & #28
- Validation : 없음

- 사용된 특징 값 : 전압, 온도 등 **물리적 변화량 21개**

→ Extract Features (d_1) : 14 x 21

데이터 기반의 상태 추정 방법론

- **1-3. Feature Extraction with Autoencoder**



< Framework에서 Feature Extraction >

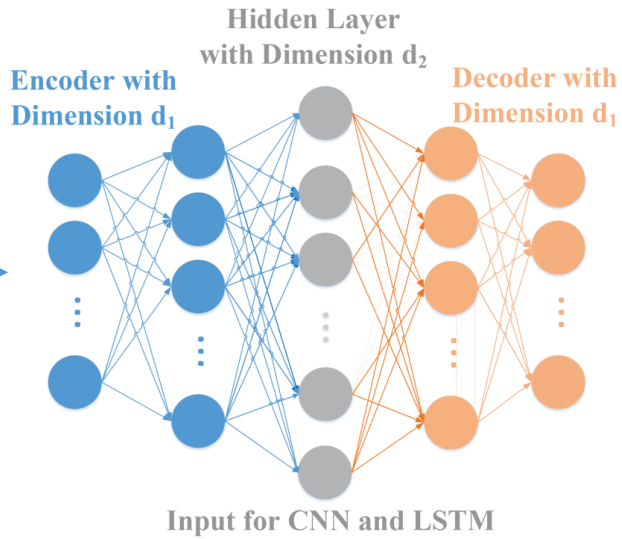


Fig. 2. Autoencoder structure in Auto-CNN-LSTM.

▫ Latent vector : d_2 (확장된 특징 값, CNN과 LSTM Input)
d_2 size : 14 x **50**

▫ Input data : d_1 (특징 값 : 전압, 전류 온도 등 측정 가능한 물리적 변수 값)
d_1 size : 14 x 21 (충전, 방전을 총 14번 수행 x 21개의 특징 값)

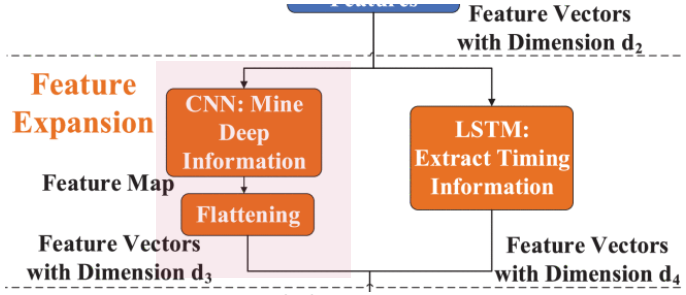
TABLE II
COMPARISON OF ACCURACY WITH DIFFERENT OUTPUT SIZES OF ENCODER

Output Size of Encoder	Filters to Smooth	*RMSE(%)*	*Accuracy(%)*
40	without filter	9.43	90.57
	Linear filter	9.32	91.68
	Second-order filter	9.39	90.61
	Third-order filter	9.39	90.61
45	without filter	8.04	91.96
	Linear filter	7.98	92.02
	Second-order filter	8.03	91.97
	Third-order filter	8.03	91.97
50	**without filter**	**5.03**	**94.97**
	Linear filter	**4.84**	**95.16**
	Second-order filter	**4.98**	**95.02**
	Third-order filter	**4.98**	**95.02**
55	without filter	7.12	92.88
	Linear filter	6.97	93.03
	Second-order filter	7.09	92.81
	Third-order filter	7.09	92.81
60	without filter	7.91	92.09
	Linear filter	7.77	92.23
	Second-order filter	7.89	92.11
	Third-order filter	7.89	92.11

The encoder outputting dimensions are set to vary from 40 to 60 and the filters are selected as linear, second-order, and third-order.

데이터 기반의 상태 추정 방법론

- **2-1. Feature Expansion with CNN**



< Framework에서 Feature Expansion >

✓ Feature map 개수 (사용자 지정)

① Conv 1 통과 → 30개
② Pooling 1 통과 → 30개
③ Conv 2 통과 → 60개
④ Pooling 2 통과 → 60개
⑤ Conv 3 통과 → 120개
⑥ Flatten 통과 → 480개

✓ Pooling : Feature map 의 개수는 그대로, 크기를 변경

▫ $Input$: d_2 (Autoencoder에서 도출한 값)
d_2 size : 14 x **50**

▫ $Output$: d_3 (CNN에서 도출한 값)
d_3 size : 1 x 480

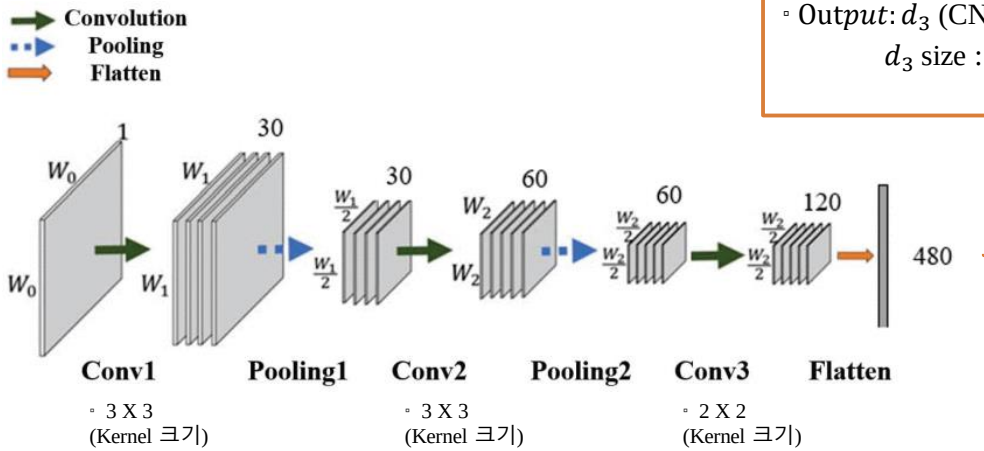
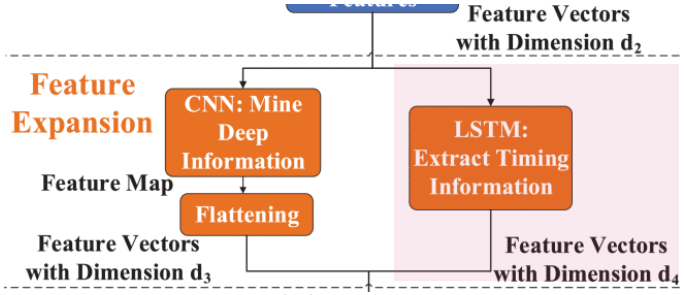


Fig. 3. CNN structure in Auto-CNN-LSTM.

데이터 기반의 상태 추정 방법론

- **2-2. Feature Expansion with LSTM**



< Framework에서 Feature Expansion >

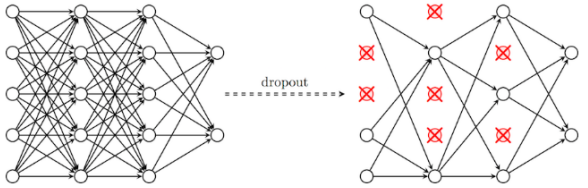
✓ LSTM 구조

: LSTM > Dropout > LSTM > Dropout > LSTM > Dropout

LSTM 에서 처리한 정보 중 일부를 잊게 하여 모델이 훈련 데이터에 과도하게 최적화되는 것을 방지

✓ Drop out

: 일부 노드를 제외시켜 Overfitting 방지



▫ *Input*: d_2 (Autoencoder에서 도출한 값)
d_2 size : 14 x **50**

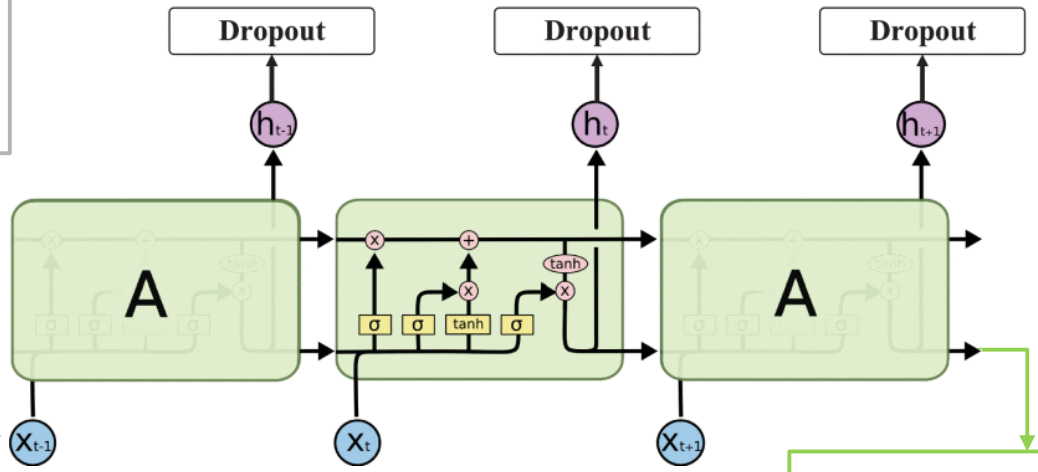
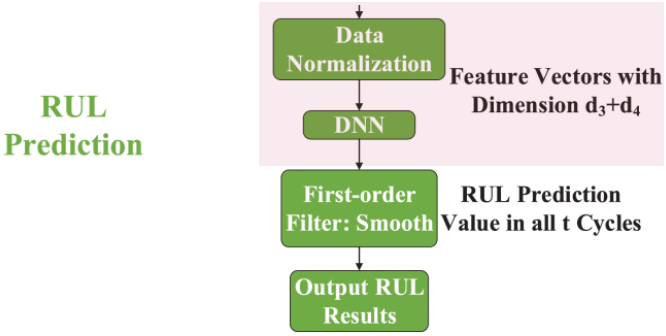


Fig. 4. LSTM structure in Auto-CNN-LSTM.

▫ *Output*: d_4 (LSTM에서 도출한 값)
d_4 size : 언급되지 않음

데이터 기반의 상태 추정 방법론

- **3. RUL Prediction with DNN**



< Framework에서 RUL Prediction>

✓ DNN 구조

: 7개의 Layer로 구성되어 있으며, Layer 간격마다 Dropout을 진행

+ L2 Regularization 을 통해 과적합을 방지

- d_3 (CNN에서 도출한 값)
  d_3 size : 1 x 480
- d_4 (LSTM에서 도출한 값)
  d_4 size : 언급되지 않음

→ d_3 & d_4 결합 : 1 x 484 → 정규화 (Min-Max)

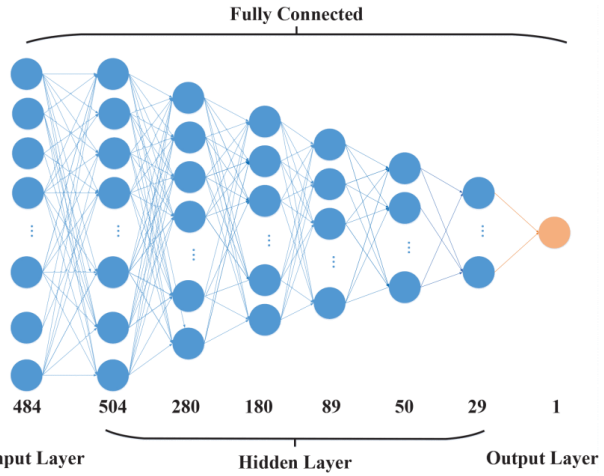
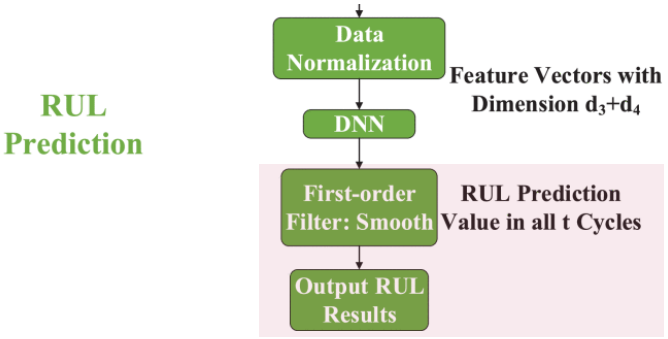


Fig. 5. DNN structure in Auto-CNN-LSTM.

→ RUL 0 ~ 1사이의 값으로 도출

데이터 기반의 상태 추정 방법론

- **3. RUL Prediction with DNN + Smoothing**



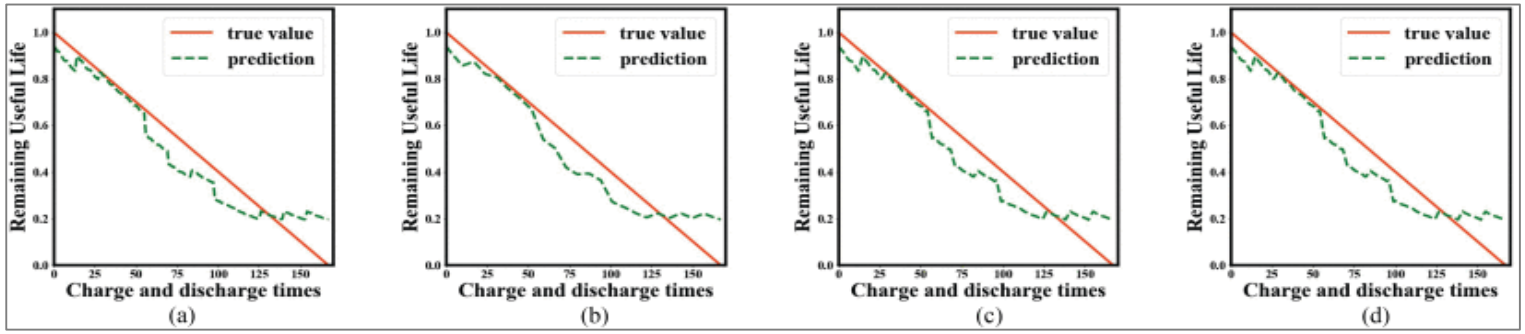
< Framework에서 RUL Prediction>

< 필터 종류에 따른 오차 값 >

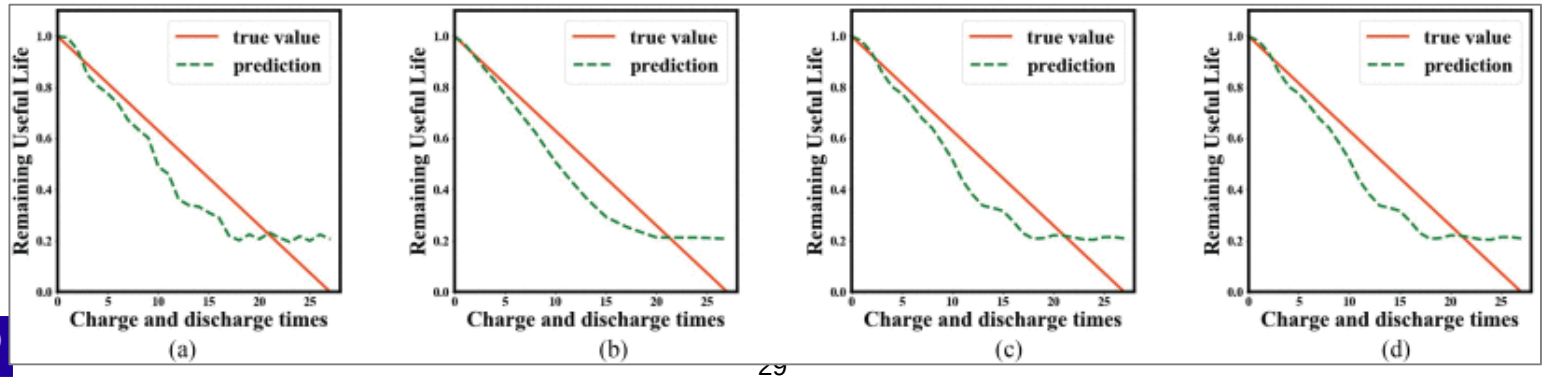
TABLE II
COMPARISON OF ACCURACY WITH DIFFERENT OUTPUT SIZES OF ENCODER

Output Size of Encoder	Filters to Smooth	*RMSE(%)*	*Accuracy(%)*
40	without filter	9.43	90.57
	Linear filter	9.32	91.68
	Second-order filter	9.39	90.61
	Third-order filter	9.39	90.61
45	without filter	8.04	91.96
	Linear filter	7.98	92.02
	Second-order filter	8.03	91.97
	Third-order filter	8.03	91.97
50	**without filter**	**5.03**	**94.97**
	Linear filter	**4.84**	**95.16**
	Second-order filter	**4.98**	**95.02**
	Third-order filter	**4.98**	**95.02**
55	without filter	7.12	92.88
	Linear filter	6.97	93.03
	Second-order filter	7.09	92.81
	Third-order filter	7.09	92.81
60	without filter	7.91	92.09
	Linear filter	7.77	92.23
	Second-order filter	7.89	92.11
	Third-order filter	7.89	92.11

The encoder outputting dimensions are set to vary from 40 to 60 and the filters are selected as linear, second-order, and third-order.



< #7 LIB(위) , #28 LIB(아래) 에 대한 필터를 사용한 Auto-CNN-LSTM의 예측 결과 (a) 필터 없음 (b) 1차 선형 필터 (c) 2차 필터 (d) 3차 필터 >

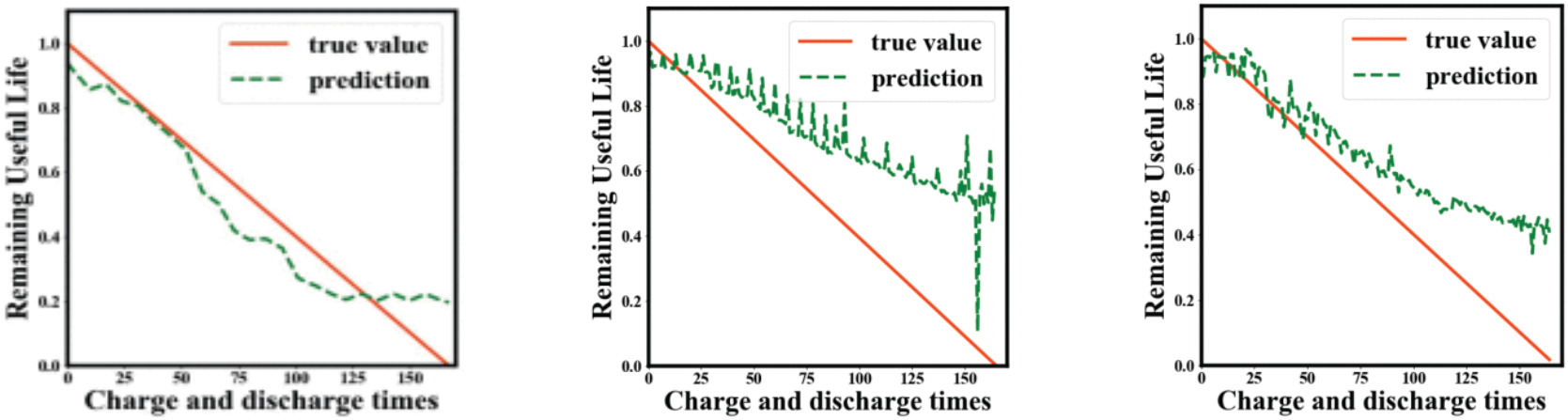


데이터 기반의 상태 추정 방법론

- 다른 방법론과 비교

Method	*RMSE(%)*	*Accuracy(%)*
Auto-CNN-LSTM	5.03	94.97
Linear Filter	**4.84**	**95.16**
Second-order Filter	4.98	95.02
Third-order Filter	4.98	95.02
ADNN	11.80	88.20
SVM	18.23	81.77

< 방법론에 따른 오차 값 >



< #7 LIB RUL 예측 결과 (a) Auto-CNN-LSTM + 1차 선형 필터 (b) Autoencoder + DNN (c) SVM >

소감 및 추후 연구 내용



- 신경망에 사용되는 용어에 대한 공부 필요
- 배터리 특성에 대한 공부가 추가로 이루어지면 더 좋을 것 같다고 생각

CVSP

감사합니다