J. Korean Inst. Electr. Electron. Mater. Eng.
Vol. 35, No. 2, pp. 134-142 March 2022
DOI: https://doi.org/10.4313/JKEM.2022.35.2.4
ISSN 1226-7945(Print), 2288-3258(Online)

Tutorial Status Report

결정그래프 합성곱 인공신경망을 통한 소재의 생성 에너지 예측

이현기, 서동화

울산과학기술원 에너지화학공학과

초록: 기존의 시행착오를 거쳐 소재를 개발하는 방법은 조금씩 한계를 보이고 있는데, 왜냐하면 산업과 기술이 고도화되고 기능성 소재가 가져야 하는 특성은 복잡해지면서 그 요구치가 높아지고 있기 때문이다. 이를 극복하기 위해 데이터 기반의 인공신경망으로 복잡한 소재 공간을 빠르게 탐색하여 소재 개발을 가속화하고자 하는 연구들이 진행되고 있다. 특히 결정그래프 합성곱 인공신경망은 결정 소재의 구조에 따른 특성을 학습하는 인공신경망으로 소재의 특성(생성 에너지, 밴드갭, 부피 탄성 계수 등)을 양자역학 기반의 제일원리 계산보다 빠르게 예측한다. 본 논문에서는 46,629개의 결정구조 데이터와 그 생성 에너지를 공공데이터베이스에서 불러와 결정그래프 합성곱 인공신경망 모델을 학습시키고 이를 특성 예측에 적용해 보는 예제를 설명한다. 이를 통해 간단한 프로그래밍 지식으로 소재 특성 예측 모델을 재현해 보고 원하는 데이터 셋과 연구 분야에 적용할 수 있을 것으로 기대된다. 인공지능 모델의 개발은 앞으로 더 복잡한 특성을 가져야만 하는 소재의 개발을 위해 넓은 범위의 소재를 탐색해야만 하는 과정을 획기적으로 단축시켜 소재 개발의 가속화를 촉진시킬 것으로 생각된다.

키워드: 결정그래프 합성곱 인공신경망, 머신러닝, 딥러닝, 인공지능, 합성곱 인공신경망

Prediction of Material's Formation Energy Using Crystal Graph Convolutional Neural Network

Hyun-Gi Lee and Dong-Hwa Seo

School of Energy and Chemical Engineering, Ulsan National Institute of Science and Technology(UNIST), Ulsan 44919, Korea

(Received January 14, 2022; Revised January 20, 2022; Accepted January 22, 2022)

Abstract: As industry and technology go through advancement, it is hard to search new materials which satisfy various standards through conventional trial-and-error based research methods. Crystal Graph Convolutional Neural Network(CGCNN) is a neural network which uses material's features as train data, and predicts the material properties(formation energy, bandgap, etc.) much faster than first-principles calculation. This report introduces how to train the CGCNN model which predicts the formation energy using open database. It is anticipated that with a simple programming skill, readers could construct a model using their data and purpose. Developing machine learning model for materials science is going to help researchers who should explore large chemical and structural space to discover materials efficiently.

Keywords: Crystal graph convolutional neural network, Machine learning, Deep learning, Artificial intelligence, Convolutional neural network

✉ **Dong-Hwa Seo; dseo@unist.ac.kr**

1. 서 론

■ 지구온난화 문제로 친환경 에너지의 사용이 강조되면서, 친환경 에너지의 생성 [1,2]과 에너지 저장 [3,4]에 대한 소재 개발에 관심이 집중되고 있다. 하지만 이러한 소재들이 부반응 없이 소재의 뛰어난 특성을 유지하게 하기 위해서는 다양한 구조와 조성에 대한 탐색이 필요하고, 기존의 시행착오를 통한 소재 개발 과정인 에디슨적 접근법 방식으로는 모든 구조와 조성에 대한 탐색이 어렵다. 무어의 법칙에 따라 프로세서의 개발이 가속화되면서 대두된 전산모사 기법은 실험적 탐색법을 줄이는 획기적인 방법으로 사용되고 있으나 [5], 구조 내 원자 수가 많아지면 계산 시간이 급증하는 문제가 있어 더 나은 탐색 방법이 필요하다.

■ 알파고의 등장으로 주목받게 된 인공신경망은 데이터를 통해서 학습된 인공지능이 많은 영역에서 사람보다 뛰어난 역할을 수행할 수 있다는 것을 증명했고, 이에 머신러닝을 통한 실험 조건 최적화 [6,7], 소재의 물성 예측 [8]과 같이 재료공학 역시 인공지능을 활용한 연구가 진행되고 있다.

■ 인공지능을 활용해 결정구조의 물성을 예측하는 모델을 만들기 위해서는 다양한 크기의 단위격자에 다양한 원자의 종류 및 개수를 가지는 결정구조를 정해진 길이의 벡터로 부호화(encoding)해야 한다. 이를 위해 대부분의 인공지능 모델은 결정이 가지는 특성을 추출하거나 [9,10], 원자 결합을 변형시켜 부호화하는데 [11], 전자의 경우는 결정구조의 특성을 추출할 때 전산모사나 분자동역학 같은 다른 계산 도구를 이용해야 하고, 후자는 부호화된 행렬이 복잡해 모델의 해석이 어렵다는 단점이 있다.

■ 결정그래프 합성곱 인공신경망(crystal graph convolutional neural network, CGCNN) [12]은 단순히 원자의 종류와 위치를 나타내는 결정 구조 파일(crystallographic information file, CIF)을 이용해 결정구조를 부호화하기에 예측을 위해서 결정구조의 특성을 다른 계산 도구를 이용해 만들거나 복잡한 변형 과정을 거치지 않는다. 또한 이미지 선별을 위한 인공지능 모델인 합성곱 인공신경망(convolutional neural network, CNN)과 동일한 방법으로 예측하기 때문에 복잡한 문제에 대해서도 범밀도함수이론(density functional theory, DFT) 계산과 비슷한 결과를 보이면서, 예측에 필요한 시간이 짧고 [13], 학습된 모델이 물질의 디자인 기준을 제공할 수 있다. 본 Tutorial Status Report에서는 CGCNN 모델의 원리 및 결정구조의 formation energy 예측 모델을 재현하는 방법을 소개한다. 이를 통해 간단한 프로그래밍 지식으로 원하는 데이터 셋과 연구 분야에 맞는 모델을 구축하고 적용해 많은 결정구조의 물성을 빠르게 스크리닝 할 수 있을 것으로 기대된다.

2. 본 문

2.1 CGCNN 메커니즘 및 결정구조 부호화

■ 그림 2(a)는 CGCNN 모델의 전체적인 메커니즘을 보여주고 있다. CGCNN은 원자에 대한 정보를 담고 있는 노드(node, 원자 벡터, v)와 결합에 대한 정보를 담고 있는 에지(edge, 결합 벡터, u)로 이뤄진다. 결정구조 내 존재하는 원소들의 정보를 원자 벡터에 부호화하기 위해 원-핫 인코딩(one-hot encoding) 방법을 사용하는데, 예를 들어 Nickel(Ni, 28번)의 경우 총 18개의 족에

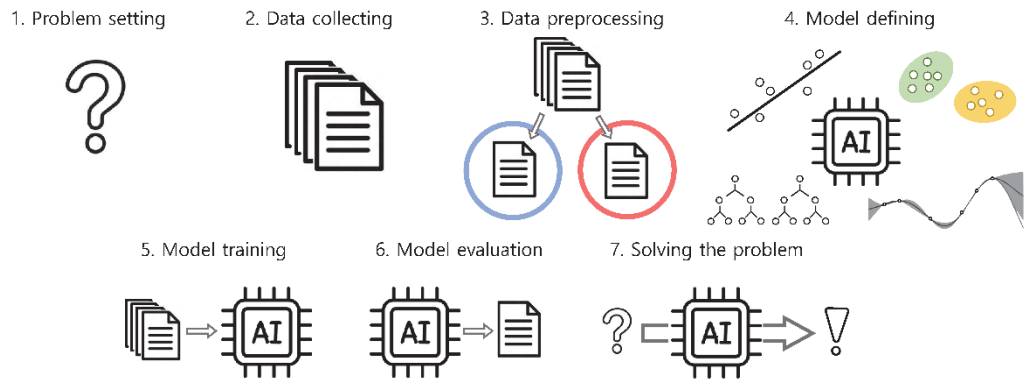

Fig. 1. Schematic diagram to show how Artificial Intelligence(AI) solve a problem.

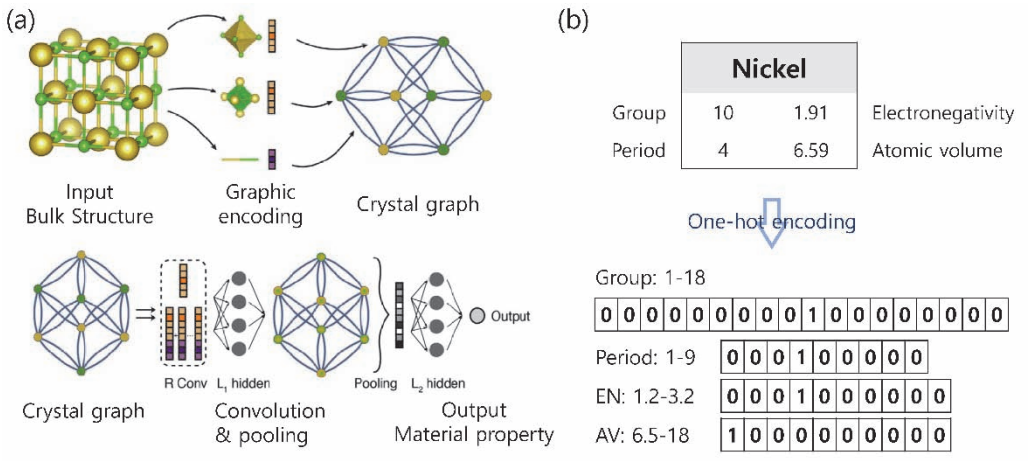

Fig. 2. (a) Schematic of Crystal Graph Convolutional Neural Network(CGCNN) . Reproduced with permission. [12] Copyright 2018, American Physical Society and (b) an example of one-hot encoding of nickel element in CGCNN model.

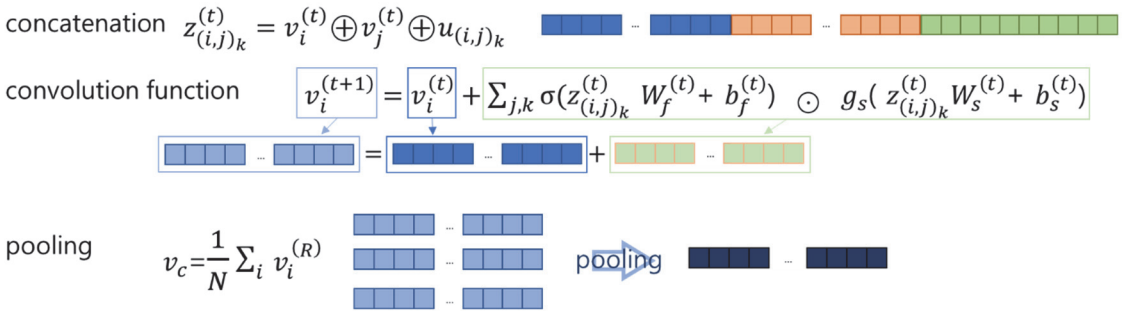

Fig. 3. Figure to show concatenation vector z, convolution layer and pooling layer.

서 족 번호가 10번, 총 9개의 주기(7주기 + 란타넘, 악티넘)에서 주기 번호가 4번이기 때문에 v_{Ni} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]으로 부호화된다. 연속적인 값을 가지는 원자의 특성과 원자 간 결합 길이의 경우 상한과 하한 사이를 동일한 간격으로 나눠 부호화한다. 이 방법으로 다른 정보 없이 결정구조의 정보를 통해 다양한 크기의 결정구조를 일정한 크기의 벡터로 부호화하는 것이 가능하다.

2.2 합성곱 레이어

■ 부호화된 원자 벡터와 결합 벡터를 통해 결정구조의 물성을 예측하는 모델을 만들기 위해서 CGCNN은 일정 거리 안에 있는 원자 두 개가 결합으로 연결되어 있다고 간주한다. 그리고 인접한 두 원자의 원자 벡터와 그 결합 벡터를 이어 붙이는 연결(concatenation) 과정을 통해 z 벡터를 만들고, 합성곱 행렬을 통과해 새로운 원자 벡터를 만든다. n번의 합성곱 레이어를 거쳐 나온 각 세대 원자 벡터를 평균 풀링(average pooling)을 통해 원자 벡터와 같은 차원의 벡터를 만들고 완전 연결 레이어(fully connected layer)를 거쳐 물성을 예측한다. 그림 3은 합성곱 함수와 풀링 함수를 나타내는 그림 및 식으로 z는 인접한 두 원자 벡터 v_i, v_j와 그 결합 벡터 $u_{(i,j)k}$를 연결한 벡터이고, W는 가중치 행렬, b는 편차 벡터, σ는 시그모이드(sigmoid) 함수, g는 Rectified Linear Unit (ReLU) 함수, ⊙은 행렬의 같은 위치의 원소의 곱(element-wise multiplication)을 의미한다.

■ 합성곱 함수에서 중요한 점은 z 벡터 안의 두 개의 원자 벡터와 하나의 본드 벡터가 각각 다른 가중치에 영

향을 받아야 한다는 것이다.

$$\boldsymbol{\nu}_i^{(t+1)} = g\left[\left(\sum_{j,k}\boldsymbol{\nu}_j^{(t)} \oplus \boldsymbol{u}_{(i,j)_k}\right)\boldsymbol{W}_c^{(t)} + \boldsymbol{\nu}_i^{(t)}\boldsymbol{W}_s^{(t)} + \boldsymbol{b}^{(t)}\right], \quad (1)$$

$$\boldsymbol{\nu}_i^{(t+1)} = \boldsymbol{\nu}_i^{(t)} + \sum_{j,k}\sigma\left(\boldsymbol{z}_{(i,j)_k}^{(t)}\boldsymbol{W}_f^{(t)} + \boldsymbol{b}_f^{(t)}\right) \odot g\left(\boldsymbol{z}_{(i,j)_k}^{(t)}\boldsymbol{W}_s^{(t)} + \boldsymbol{b}_s^{(t)}\right) \quad (2)$$

식 (2)의 경우 각각의 벡터에 다른 가중치가 할당되어 벡터의 특성을 더 정확히 반영해 학습되기 때문에 높은 명중률의 모델을 만들 수 있다. 식 (1)처럼 하나의 원자만 고려해서 만든 합성곱 함수을 사용하면, 연결된 다른 원자와 그 원자와의 결합을 따로 고려하지 않기 때문에 모델의 정확도가 크게 감소한다.

2.3 CGCNN 레이어의 종류

■ 인공신경망의 학습에서 고려해야 할 부분은 모델이 가지는 파라미터의 개수다. 파라미터는 가중치 행렬(W)과 편차 벡터(b)를 의미하고 인공신경망의 학습 중에 수정되어 모델의 명중률을 향상시킨다. 파라미터의 개수가 많다면 복잡한 문제에 대해 해결 능력이 오를 수 있지만 더 많은 학습 데이터가 필요하고, 적다면 학습 데이터의 개수는 적게 필요하지만 복잡한 문제에 대한 해결 능력이 떨어진다. CGCNN 모델 내 파라미터의 개수는 레이어의 차원 및 개수를 바꾸는 방법으로 조절이 가능한데 그림 4는 CGCNN 모델에서 사용되는 레이어의 종류와 각 레이어 내에 존재하는 파라미터들을 설명하였다.

■ 부호화된 벡터들이 모델 안에 들어가면 우선 선형 변환을 통해 벡터의 차원을 정해진 차원으로 맞춘다. 이는 사람마다 사용하는 원자 특성의 개수가 다를 수 있기 때문에 다양한 길이의 원자 벡터가 생성될 수 있는데, 긴 원자 벡터가 변환 없이 반영되면 이후에 진행될 합성곱 레이어에서 너무 많은 파라미터가 생성되어 학습이 어렵기 때문이다. 이후 두 원자 벡터와 그 결합 벡터를 연결해 z 벡터를 만들고, 지정된 횟수만큼 합성곱 레이어를 진행한다. 합성곱 레이어의 결과로 생성된 각 세대의 결과물 벡터를 평균 풀링을 통해 원래 원자 벡터와 똑같은 길이의 벡터를 만든다. 풀링은 마지막 합성곱 레이어의 결과물만을 고려하는 게 아니라 각 합성곱 레이어의 결과물들을 전부 평균을 취하는데, 이렇게 전 세대를 고려하는 이유는 합성곱 레이어를 거치지 않은 원자 벡터를 반영하면 모델의 정확도가 올라가는 경우가 많기 때문이다. 이후 hidden feature length 길이로 벡터를 변환하기 위해 선형 변환을 한 번 더 진행하고, 일반적인 인공신경망 레이어를 정해진 횟수만큼 진행한다. 이후 벡터를 스칼라로 변환해 원하는 물성의 예측을 도출한다.

■ CGCNN 모델에서 레이어의 종류 및 개수 외에 중요한 변수는 사용하고자 하는 원자 특성의 개수이다. 필요한 원자 특성의 개수는 예측하고자 하는 물성마다 다 다르며, 너무 많게 되면 과적합(overfitting)이 일어나 명중률이 낮아지고, 너무 작으면 복잡한 문제에 대한 해결 능력이 떨어져 명중률이 낮아지기 때문에 문제마다 최적화가 필요하다.

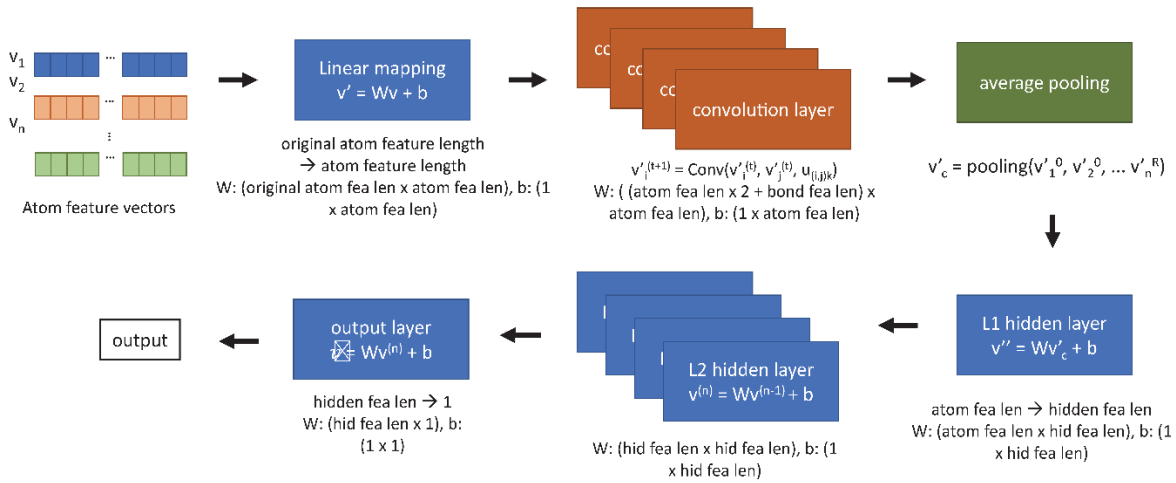

Fig. 4. Type of layers and their number in CGCNN model and dimension of weight and bias in the layers.

2.4 CGCNN 모델 구축, 학습 및 평가

■ 본 Tutorial Status Report에서는 인공지능을 통한 소재 개발의 선구자이신 Jeffry C. Grossman 연구실의 Tian Xie 박사가 만든 CGCNN 파이썬 패키지 [12]를 이용해 결정구조의 원자 당 생성 에너지를 예측하는 모델을 재현하는 방법을 설명한다. 이를 위해 학습, 검증, 평가 데이터로 Materials Project(MP)의 결정구조 데이터 46,629개를 사용했고, 대량의 데이터를 많은 epoch로 반복학습하기 위해 구글이 제공하는 google colab의 연산 자원을 활용했다.

■ CGCNN 패키지에는 모델의 예측을 위한 main.py, 결정구조 데이터를 불러와 벡터로 저장하는 data.py, 합성곱 함수 같은 레이어에 대한 정보가 pytorch 패키지를 사용해 만들어진 model.py, 그리고 완성된 모델을 이용해 물성을 예측하는 predict.py로 구성되어 있다. 모델을 학습시키기 위한 데이터는 원자 특성을 원-핫 인코딩한 atom_init.json, mp-id와 소재의 물성 정보를 저장한 id_prop.csv, 결정구조의 정보인 mp-id.cif로 구성되어 있다.

■ CGCNN 패키지의 작동 방식은 1) 경로 및 하이퍼 파라미터 설정 후 main.py 실행 2) data.py가 정해준 경로에 접근, cif 파일의 결정구조를 atom_init.json에 따라 부호화, id_prop.csv에 따라 물성과 결정구조를 매칭 3) main.py는 부호화된 데이터와 하이퍼 파라미터를 model.py로 전송 4) model.py는 모델을 구축, 정해진 epoch 횟수만큼 학습을 진행 5) 가장 명중률이 높았던 모델을 평가 데이터로 평가, 이후 결과를 저장하는 단계로 진행된다. 출력 파일은 평가 데이터의 예측 결과를 저장한 test_result.csv, 반복학습하면서 가장 좋았던 모델을 백업한 checkpoint.pth.tar, 학습 중 가장 좋은 모델을 저장한 model_best.pth.tar로 이뤄진다. 그림 5(a), (b)에는 CGCNN의 디렉토리 구성 및 작동 메커니즘을 표현하였다. 예시에서는 모델의 재현을 위해 epoch = 1,000, lr = 0.02, lr-mile = 800, h-fea-len = 32, n-conv = 4로 하이퍼 파라미터를 사용했다. 데이터를 학습:검증:평가 = 6:2:2로 나누었으며 colab의 tesla P-100 gpu 사용 시 총 3시간 46분의 시간이 소요되었다.

2.5 CGCNN 모델 구축 예제

■ 본 장에서는 MP 결정구조 데이터를 가져와 전처리 후 CGCNN 모델을 학습하여 결정 소재 생성 에너지 예측에 적용하는 과정을 순차적으로 설명하고자 한다. 본 예제에 대한 코드는 https://github.com/Hyun-Gi/kieeme_tutorial_CGCNN에서 다운로드 받아 활용할 수 있다.

2.5.1 Materials Project에 등록 및 API key 획득

제일원리 계산 데이터베이스 Materials Project(MP)의 Application Programming Interface(API) key를 받는다. 학습을 위한 다량의 결정구조 데이터를 불러올 때는 API를 사용해 조건을 주어 불러오는 것이 효과적이다 [14].

2.5.2 구글 계정을 이용해 google colab에 접속 및 런타임 유형을 GPU로 변경

구글 계정을 이용해 colab에 접속한다. Google은 머신러닝을 처음 접하는 사람들을 위해 연산 자원을 일정량 무료로 제공한다. 다량의 데이터를 학습하기 때문에 런타임 유형을 GPU로 바꾼다.

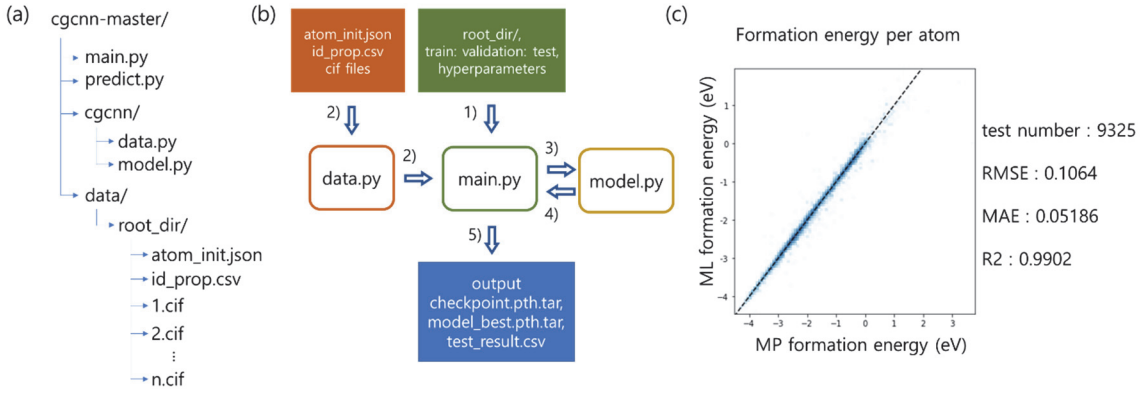

Fig. 5. (a) Configuration of CGCNN directory, (b) a flowchart of CGCNN, and (c) a result of CGCNN model predicting material's formation energy per atom.

2.5.3 MP API를 이용해 결정구조의 mp-id, formation energy, cif 저장

문제의 조건에 맞는 데이터를 불러온다. 이 예제에서는 formation energy를 예측하는 모델을 재현하기 위해 CGCNN 패키지의 material-data 폴더 내 mp-ids-46744.csv의 mp-id에서 formation energy 정보가 누락된 115개의 데이터를 제거한 46,629개의 데이터에 대한 cif 파일들과 mp-id 및 formation energy를 담은 csv 파일을 만들어 사용했다. 만약 colab을 사용한다면 데이터가 너무 많아 업로드 시 데이터 누락이 발생하는데, 모든

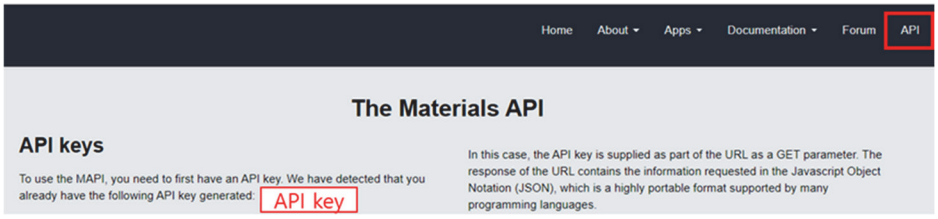

Fig. 6. Getting API from MP.

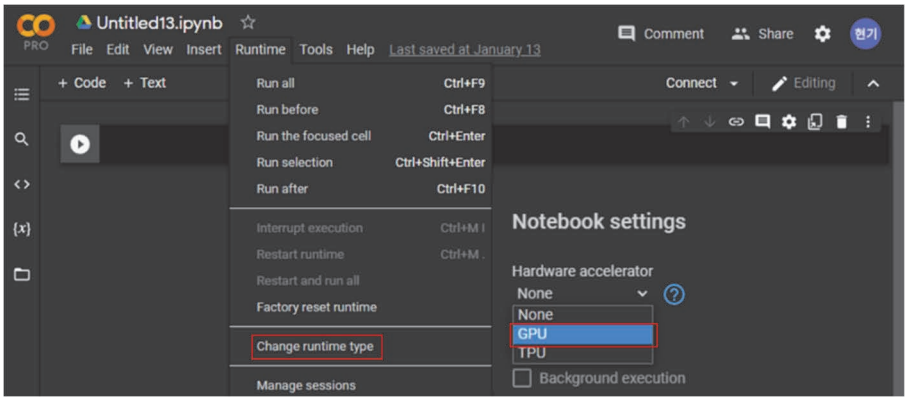

Fig. 7. Setting runtime type to GPU option to accelerate computation.

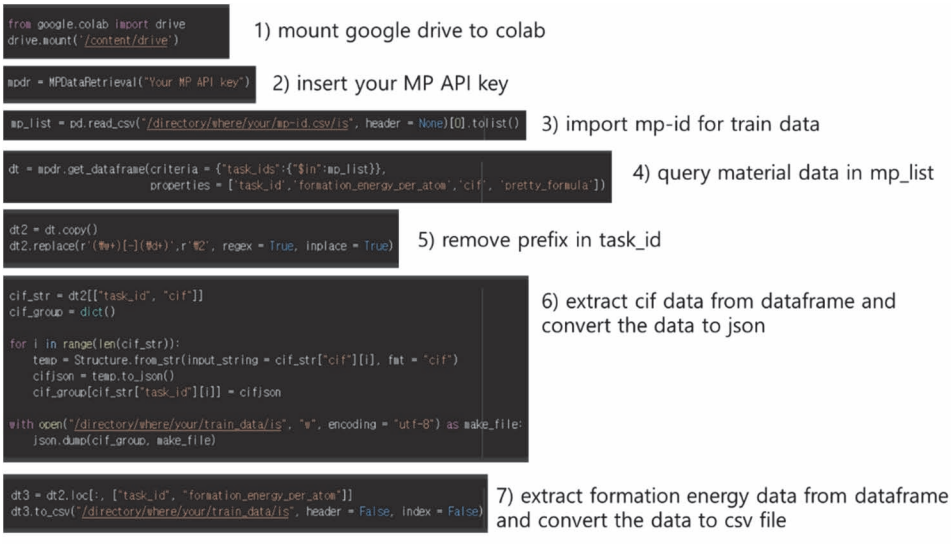

Fig. 8. Collecting formation energy data from MP and preprocessing of the data.

cif 파일들을 json 파일로 dump 시키면 누락 없이 데이터를 학습시킬 수 있다. atom_init.json 파일은 data/sample-regression에서 가져온다.

2.5.4 CGCNN 패키지 다운로드

Github에서 CGCNN 패키지를 받는다. 패키지를 사용하기 위해서는 Pytorch, scikit-learn, pymatgen이 구동환경에 설치되어 있어야 한다. cif 파일들을 json으로 데이터를 받았으면 패키지 내 data.py를 일부 수정해 json 파일을 읽을 수 있도록 바꾼다.

2.5.5 하이퍼 파라미터 설정 후 모델 학습 시작

하이퍼 파라미터 설정 후 main.py를 시작한다. 모델이 지정된 batch 수만큼 학습할 때마다 현재 모델의 MAE를 출력하는데 이 결과가 validation의 MAE와 차이가 크다면 과적합 문제일 가능성이 있으므로 하이퍼 파라미터를 수정해 과적합을 해결해야 한다.

2.5.6 모델 결과를 분석

test_result.csv 파일에서 모델의 예측 성능을 확인한다.

```
        ###############################################################
        # import cif_group.json
        cif_json_file = os.path.join(self.root_dir, 'cif_group.json')
        print("cif_json directory :", cif_json_file)
        assert os.path.exists(cif_json_file), 'cif_group.json does not exist!'
        with open(cif_json_file, 'r') as read_file:
            self.json_data = json.load(read_file)
        ###############################################################

    def __len__(self):
        return len(self.id_prop_data)

    @functools.lru_cache(maxsize=None)  # Cache loaded structures
    def __getitem__(self, idx):
        cif_id, target = self.id_prop_data[idx]
#         crystal = Structure.from_file(os.path.join(self.root_dir, cif_id+'.cif'))
        ###################################################################
        # extract cif data from json file
        crystal = Structure.from_str(self.json_data[cif_id], fmt = "json")
        ###################################################################
```

Fig. 9. Modifying data.py in CGCNN package to read json file containing cif files.

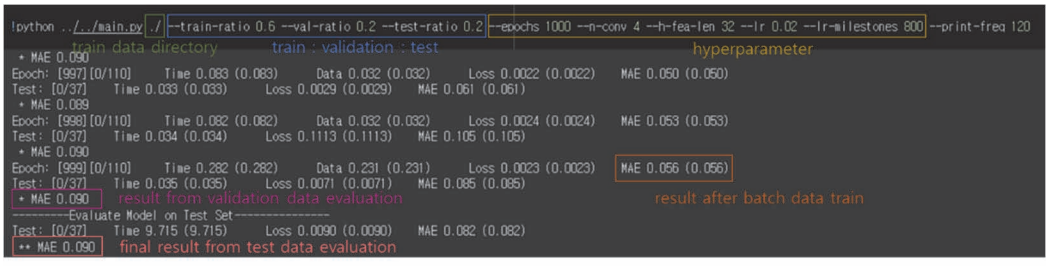

Fig. 10. Training model with the hyperparameter and result of model training.

```
dt = pd.read_csv("/directory/where/yourt/test_result.csv/is", header = None)
predicted_data = list(dt.iloc[:,2])
real_data = list(dt.iloc[:,1])

test_number = len(predicted_data)

RMSE = mean_squared_error(real_data, predicted_data, squared = False)
MAE = mean_absolute_error(real_data, predicted_data)
r2 = r2_score(real_data, predicted_data)
```

Fig. 11. Plotting result from test_result.csv which is made from testing data.

2.6 설명가능한 인공지능

■ 원래 학습된 인공지능의 작동 과정은 다양한 레이어 안의 가중치들에 영향을 받기 때문에 인간이 이해할 수 없다. 그렇기에 CNN 모델에서는 이해 가능한 인공지능(explainable artificial intelligence, XAI)을 구축하기 위해 많은 연구가 이뤄졌는데, CGCNN 모델도 결정 구조를 부호화한 CNN이기 때문에 이러한 아이디어를 모델에 반영할 수 있다. 예를 들어 CAM [15]은 이미지 분류를 위한 CNN에서 모델의 레이어가 출력하는 결과물을 매핑(mapping)한 것으로, 이를 통해 실제 인공신경망이 입력값을 받았을 때 어떻게 출력값을 내는지, 중요하게 고려하는 부분이 무엇인지를 확인할 수 있다 [그림 12(a), (b)]. 이와 비슷한 방법으로 CGCNN의 말단 레이어의 차원을 줄여 원자 벡터만 추출한 뒤 소재의 물성을 예측하면 마지막 레이어의 정보로부터 원자들이 물성에 어느 정도 영향을 미치는지 설명 가능한 모델을 구축할 수 있다. Grossman 연구실에서는 페로브스카이트 구조(ABX_3) 18,928개의 데이터를 통해 energy above hull을 예측하는 모델을 구축했고, energy above hull을 낮추기 위해서 A site와 B site에 들어가는 원자들의 종류를 모델 해석을 통해 확인했다 [그림 12(c), (d)]. 그리고 이 기준으로 디자인한 구조의 상안정성을 계산한 결과 33개의 합성 가능한 소재들을 발견했다 [12]. 이처럼 데이터를 통한 모델의 학습으로, 소재 디자인의 기준을 제시할 수 있다.

3. 결 론

■ 많은 논문에서 실험에 대한 결과를 입증하기 위해, 더 나아가 실험 전 소재를 디자인하기 위해 제일원리 계산을 활용하고 있다. 제일원리 계산은 실험을 직접적으로 진행하지 않아도 되기에 시간적, 비용적 자원을 아낄 수 있지만, 실험 조건을 전부 고려하지 못하기 때문에 계산과 실험 결과의 괴리는 해결하기 어렵다. 하지만 인공지능을 통한 물성 예측은 양질의 실험 데이터가 충분하다면 실험 결과와 비슷한 결과를 예측할 것이라 기대되기 때문에 이런 괴리를 해소할 수 있다.

■ 본 tutorial report에서는 CGCNN의 생성 에너지 예측 모델을 재현하는데 그쳤지만, CGCNN은 소재의 물성 예측 외에 다른 방향으로도 개발되어 활용될 수 있다.

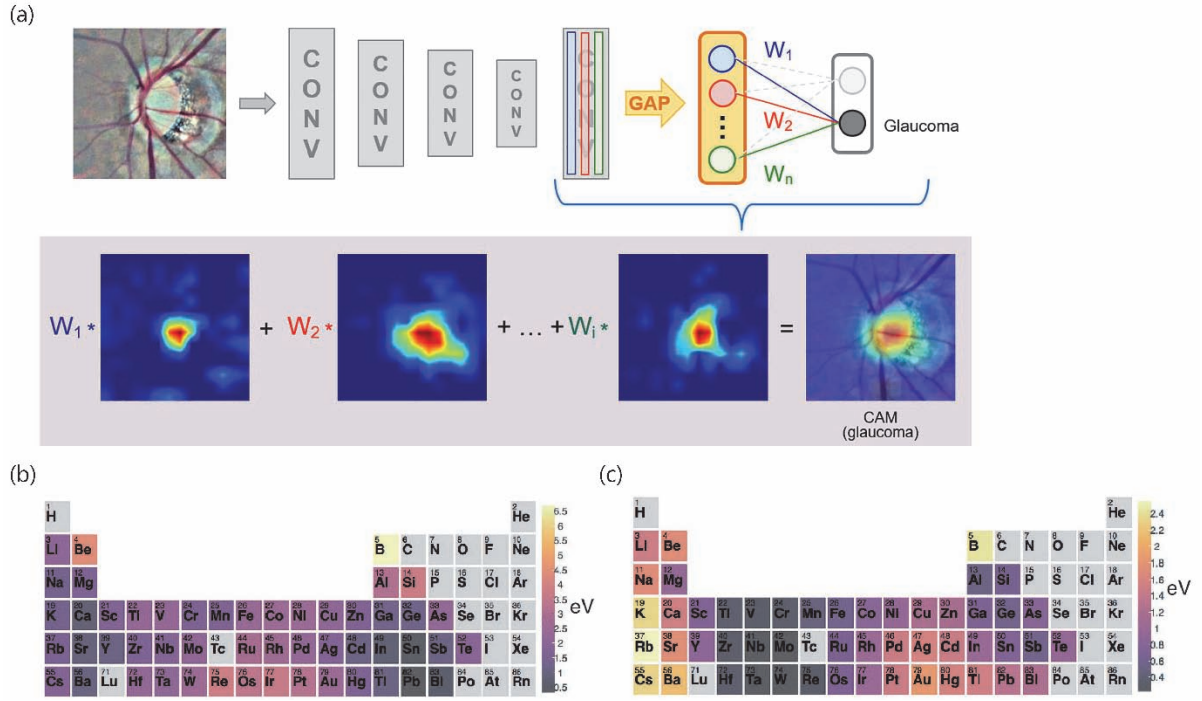

Fig. 12. (a) Class activation map attached to the final layer of the proposed CNN model. Licensed under CC BY 4.0 [15,16] (b) effect on energy above hull by A site and (c) B site element in perovskite structure (ABX_3). Reproduced with permission. [12] Copyright 2018, American Physical Society.

예를 들어, 국내 연구진들에 의해 개발된 CGCNN과 주성분 분석(primary component analysis, PCA)을 활용한 나노입자(nanoparticle)의 상태 밀도(density of state, DOS) 예측 모델 [13]은, 백금과 금 단일 원소 나노입자의 데이터를 학습해 백금-금 합금의 상태 밀도를 예측할 수 있다는 것을 보여주었다. 이 모델은 CGCNN이 하나의 물성 예측이 아니라 여러 데이터로 이뤄진 그래프를 예측할 수 있다는 것을 보여주고, 분광학(spectroscopy) 데이터 예측에도 활용될 수 있다는 가능성을 제시한다. 이렇듯 소재 개발을 위한 인공지능 모델은 추후 더 복잡한 기능을 가진 소재를 개발함에 있어 많은 후보군을 탐색할 때 실험과 계산보다 월등히 빠른 속도로 문제 해결에 필요한 소재의 특성을 예측하기에 탐색 효율을 향상시키고 소재 개발의 가속화에 기여할 것이라 기대된다.

ORCID

Dong-Hwa Seo https://orcid.org/0000-0002-7200-7186

REFERENCES

[1] H. Min, D. Y. Lee, J. Kim, G. Kim, K. S. Lee, J. Kim, M. J. Paik, Y. K. Kim, K. S. Kim, M. G. Kim, T. J. Shin, and S. Il Seok, *Nature*, **598**, 444 (2021). [DOI: https://doi.org/10.1038/s41586-021-03964-8]

[2] B. wook Park, H. W. Kwon, Y. Lee, D. Y. Lee, M. G. Kim, G. Kim, K. jeong Kim, Y. K. Kim, J. Im, T. J. Shin, and S. Il Seok, *Nat. Energy*, **6**, 848 (2021). [DOI: https://doi.org/10.1038/s41560-021-00844-3]

[3] M. Yoon, Y. Dong, J. Hwang, J. Sung, H. Cha, K. Ahn, Y. Huang, S. J. Kang, J. Li, and J. Cho, *Nat. Energy*, **6**, 362 (2021). [DOI: https://doi.org/10.1038/s41560-021-00782-0]

[4] Z. L. Xu, J. Park, J. Wang, H. Moon, G. Yoon, J. Lim, Y. J. Ko, S. P. Cho, S. Y. Lee, and K. Kang, *Nat. Commun.*, **12**, 3369 (2021). [DOI: https://doi.org/10.1038/s41467-021-23703-x]

[5] S. B. Ma, H. J. Kwon, M. Kim, S. M. Bak, H. Lee, S. N. Ehrlich, J. J. Cho, D. Im, and D. H. Seo, *Adv. Energy Mater.*, **10**, 2001767 (2020). [DOI: https://doi.org/10.1002/aenm.202001767]

[6] F. Häse, L. M. Roch, C. Kreisbeck, and A. Aspuru-Guzik, *ACS Cent. Sci.*, **4**, 1134 (2018). [DOI: https://doi.org/10.1021/acscentsci.8b00307]

[7] B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams, and A. G. Doyle, *Nature*, **590**, 89 (2021). [DOI: https://doi.org/10.1038/s41586-021-03213-y]

[8] A. D. Sendek, E. D. Cubuk, E. R. Antoniuk, G. Cheon, Y. Cui, and E. J. Reed, *Chem. Mater.*, **31**, 342 (2019). [DOI: https://doi.org/10.1021/acs.chemmater.8b03272]

[9] A. Seko, A. Togo, H. Hayashi, K. Tsuda, L. Chaput, and I. Tanaka, *Phys. Rev. Lett.*, **115**, 205901 (2015). [DOI: https://doi.org/10.1103/Phys RevLett.115.205901]

[10] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, *Nat. Commun.*, **7**, 11241 (2016). [DOI: https://doi.org/10.1038/ncomms11241]

[11] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, *Phys. Rev. B*, **89**, 205118 (2014). [DOI: https://doi.org/10.1103/PhysRevB.89.205118]

[12] T. Xie and J. C. Grossman, *Phys. Rev. Lett.*, **120**, 145301 (2018). [DOI: https://doi.org/10.1103/PhysRevLett.120.145301]

[13] K. Bang, B. C. Yeo, D. Kim, S. S. Han, and H. M. Lee, *Sci. Rep.*, **11**, 11604 (2021). [DOI: https://doi.org/10.1038/s41598-021-91068-8]

[14] S. P. Ong, S. Cholia, A. Jain, M. Brafman, D. Gunter, G. Ceder, and K. A. Persson, *Comput. Mater. Sci.*, **97**, 209 (2015). [DOI: https://doi.org/10.1016/j.commatsci.2014.10.037]

[15] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba *Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, Las Vegas, USA, 2016) p. 2921. [DOI: https://doi.org/10.1109/cvpr.2016.319]

[16] Y.C. Ko, S.Y. Wey, W.T. Chen, Y.F. Chang, M.J. Chen, S.H. Chiou, C.J.L. Liu, C.Y. Lee, *PLoS ONE*, **15** (2020). [DOI: https://doi.org/10.1371/journal.pone.0233079]