CS182 Project Overview

Yunyeong Choi[1] **Xin Chen**[1] **Jinho Shin**[1] **Aryan Chakrabarti**[1]
[1]UC Berkeley {yychoi94,chenxin0210,jhs0640,aryanc}@berkeley.edu

1 Project outlines

Predicting material properties exclusively from atomic information has been a long-standing aspiration for physicists. The development of Density Functional Theory [1] and subsequent algorithms have enabled the calculation of relatively accurate properties. However, due to the constraints of the self-consistent calculation method, the time consumption increases faster than $O(N^3)$, where N is the number of atoms. Since 2007, after J. Behler published the first Machine Learning approach [2], researchers have focused on building large databases of computed materials properties, and many different models have been tried to achieve higher accuracy with adequate computation time. In this project, we will focus on one of the most impactful papers [3], which first adapted the graph-convolution idea to predict materials properties.

The key idea of this project is that inorganic materials have many properties that can be effectively modeled using Convolutional Neural Networks (CNNs). For instance:

1. **Invariance**: As we have seen in previous work, CNNs have an inductive bias that captures translational and rotational invariance of images. In inorganic materials, symmetry operations are always used to define the material [4], and they affect the material's properties as well. In this context, one can expect that a graph-convolution network can capture symmetry with an inductive bias and perform better than other NNs.
2. **Locality**: Among the various interactions between atoms, physicists usually regard nearest-neighbor interaction as the most important and largest interaction. Except for some unique cases, this is considered true due to the form of the electrostatic potential, which is linear to $O(1/r^2)$. CNNs are an excellent choice for capturing the locality of data, and by using several CNN layers, we can also capture long-range interactions of atoms with enlarged receptive fields.
3. **Weight Sharing**: In machine learning approaches such as fully connected networks, at least 10,000 weights are typically used to predict energy. This is not ideal, since we know that the material's property is only determined by a quantum mechanical equation. To mimic this, we need to limit the number of weights and share them between different atoms. This is exactly what CNNs do.

Therefore by implementing this paper as a homework set, we not only expect to enforce student's understanding on graph-convlution network, but also expect to broaden their perspective by looking at how ML approache used in other domains.

The goal of this project is to:

1. Implement the CGCNN (Crystal Graph Convolution Neural Network) with better explanation, make a homework notebook set to understand and execute it easily.
2. Train it with a limited amount of data (focusing on only one type of material, such as perovskite.) We expect to train on $\sim$10,000 different materials, which can be done in hours with NVIDIA A100. Students will use pre-trained weights and train with smaller dataset
3. Use the trained model to predict new materials with a target property (band gap, etc...)
4. Identify the limitations of the CGCNN (such as periodicity and data augmentation) and suggest ways to improve it. (If time allows)

2 Project details

2.1 Encoding Crystal into crystal graph

The key idea of CGCNN is to represent crystal structure by a crystal graph, which encodes not only the atomic information but also bonding information between atoms based on the distance. Here we will let each atomic feature vector as $\mathbf{v}_i$, which encodes the property of the atom corresponding to node i. The edge feature vector is represented as $\mathbf{u}_{(i,j)_k}$, which encodes k-th bond connecting atom i, j. Note that there can be several bonds between atoms, which originate from the periodic nature of crystals.
Obviously, there are many ways to encode atomic and bond data. For the sake of simplicity, we will follow the methods used in the CGCNN paper, which used a pre-defined encoding vector to change atoms into vectors. These vectors have 92 dimensions, and for 100 atoms in the periodic table, they have different encodings which consist of 0 and 1. For the edge feature vectors, there also can be many ways to encode bond information; bond length, angle, and covalency. However, we will only use bond-length information between nearest neighbors, by applying a Gaussian kernel to change it to encoding vectors. One can follow the homework notebook on how this embedding is done to NaCl(salt) Crystal.
Note that this encoding could be very simple to deliver all information of interactions in crystal. Recent studies also focused on improving this; Some consider angles to capture three-body interaction, or even Mol2vec, which is a molecule version of Word2vec, is developed to change encode effectively.

2.2 Convolutional graph neural network

After building the crystal graph, the convolutional neural network act on top of the graph, and consecutive fully connected networks and pooling layer are used to predict the property of crystals. The convolution layer iteratively updates the atom feature vector $\mathbf{v}_i$ by message passing with surrounding atoms and bonds with a non-linear graph convolution function. It can be written as Eq 1.

$$v_i^{(t+1)} = Conv(v_i^{(t)}, v_j^{(t)}, u_{(i,j)_k}), (i,j)_k \in G \tag{1}$$

After iterations of convolution layers, the pooling layer of Eq 2 is used to produce overall feature vector $\mathbf{v}_c$ for the crystal, and consecutive two fully connected hidden layers are used to generate outputs. The overall architecture is not only confined to the regression problem but also can be used for classification. In the case of regression, the error function is set to MSE, and in the case of classification, NLL (Negative Log Likelihood) loss is used.

$$v_c = Pool(v_0^{(0)}, v_1^{(0)}, ..., v_N^{(0)}, ..., v_N^{(R)}) \tag{2}$$

Note that this 'convolutional' graph is not exactly meaning the CNN; it is doing message passing in the crystal graph we built. Therefore, if we maintain the permutation invariance of the convolution layer we can choose any function we want. In the original paper, the following Eq 3 and Eq 4 are tested.

$$v_i^{(t+1)} = g[(\sum_{j,k} v_j^{(t)} \oplus u_{(i,j)_k})W_c^{(t)} + v_i^{(t)}W_s^{(t)} + b^{(t)}] \tag{3}$$

$$v_i^{(t+1)} = v_i^{(t+1)} + \sum_{j,k} \sigma(z_{(i,j)_k}^{(t)} W_f^{(t)} + b_f^{(t)}) \odot g(z_{(i,j)_k}^{(t)} W_s^{(t)} + b_s^{(t)}) \tag{4}$$

where $\oplus$ denotes the concatenation of vectors, $\odot$ denotes element-wise multiplication, and $W_c^{(t)}, W_s^{(t)}, b^{(t)}$ are the convolution weight matrix, self-weight matrix, and bias. g is the activation function and σ is sigmoid function. Note that Eq 3 weight matrix $W_c^{(t)}$ is shared between all different neighbors, which is not ideal to describe materials property and results in a large validation error. The author used Eq 4 instead of that, which differentiates interactions between neighbors and added residual connection to prevent gradient vanishing.
In this project, we will change the convolution layers to compare and analyze which type of convolution layer is the best to describe the property of crystals. Students will test following three different convolution layers and compare their training/validation errors of them;

1. No convolution, only a self-weight matrix for each node.
2. Shared convolution weight matrix between different sets of nodes.
3. Different convolution weight matrices between different sets of nodes.

2.3 Dataset

Ideally, this architecture can be used to predict any property of materials. However, for simplicity, we will limit this to predicting energy and simple classification of materials family. For the homework dataset, 700 structures of the Perovskite family in the Materials Project Database are chosen. Also, 1000 oxide materials are randomly chosen for the purpose of the classification test. There are two different kinds of training; First, one will use to train only with 700 perovskite structures, and going to predict the energy of materials. Second, students will use a mixed data set of perovskite and random oxide materials, which has classifications of 1 and 0. One will use it to train the classification network and going to check whether this network can classify perovskite from other oxide materials. Note that we only mix oxides to the perovskite dataset since most perovskite includes oxygen, so if mixed with random materials, it can bypass the architecture by only training weights related to the oxygen atom encoding vector.

References

[1] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, Nov 1965.

[2] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98:146401, Apr 2007.

[3] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120:145301, Apr 2018.

[4] Wikipedia. Space group — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Space%20group&oldid=1119183693`, 2023. [Online; accessed 23-March-2023].