Predicting Molecular Properties

Group member: Yan Gao
Instructor: Pooran Singh Negi

Research Questions

Small graph problem

Some particular large graph could be devided into many small connected components.

How to scale the influence of a small graph to each node in it?

How to build a efficient graph model to scale those features and feed them to machine learning model?

Model: Deep learning model with keras

Input: Numeric and categorical features

Output: Continuous response

Molecule	FaceBook Group
molecule name	unique group name
atom index	each person's id in the group
what kind of atom	the social title of each person
hidden atom	who joined the group but with unpublic personal infos even could be a net police
spatial position of each atom	the group title of each person
coupling type	the social titles of each pair of group members and their relation(stranger, friends, enemy)
distance between atoms	how close each pair of group members look like
coupling constant(our target)	how truly close are the relations among each pair of group members

Dataset – Molecular Properties

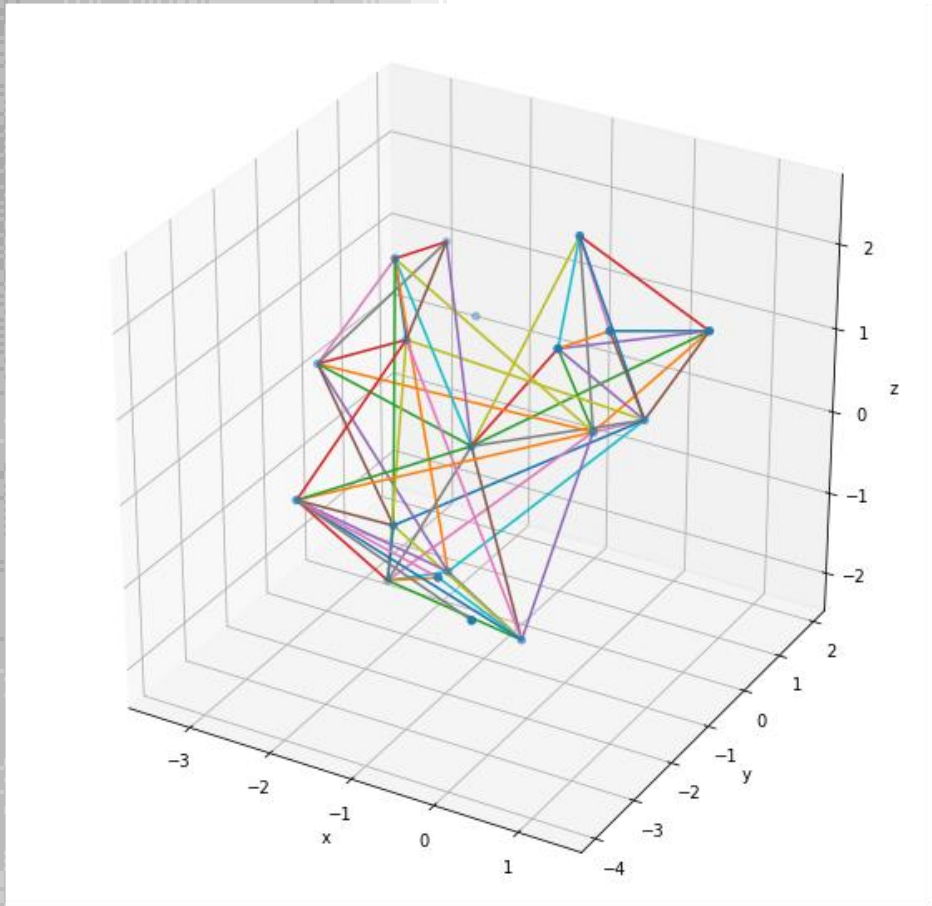
This is a Kaggle competition dataset. This is a typical relational dataset. The train and test file contains the pairs relation of atoms in different molecules. There are 85003 molecules in train data and each of them contain 5 to 30 atoms.

	id	molecule_name	atom_index_0	atom_index_1	type	scalar_coupling_constant
0	0	dsgdb9nsd_000001	1	0	1JHC	84.8076
1	1	dsgdb9nsd_000001	1	2	2JHH	-11.2570
2	2	dsgdb9nsd_000001	1	3	2JHH	-11.2548
3	3	dsgdb9nsd_000001	1	4	2JHH	-11.2543
4	4	dsgdb9nsd_000001	2	0	1JHC	84.8074

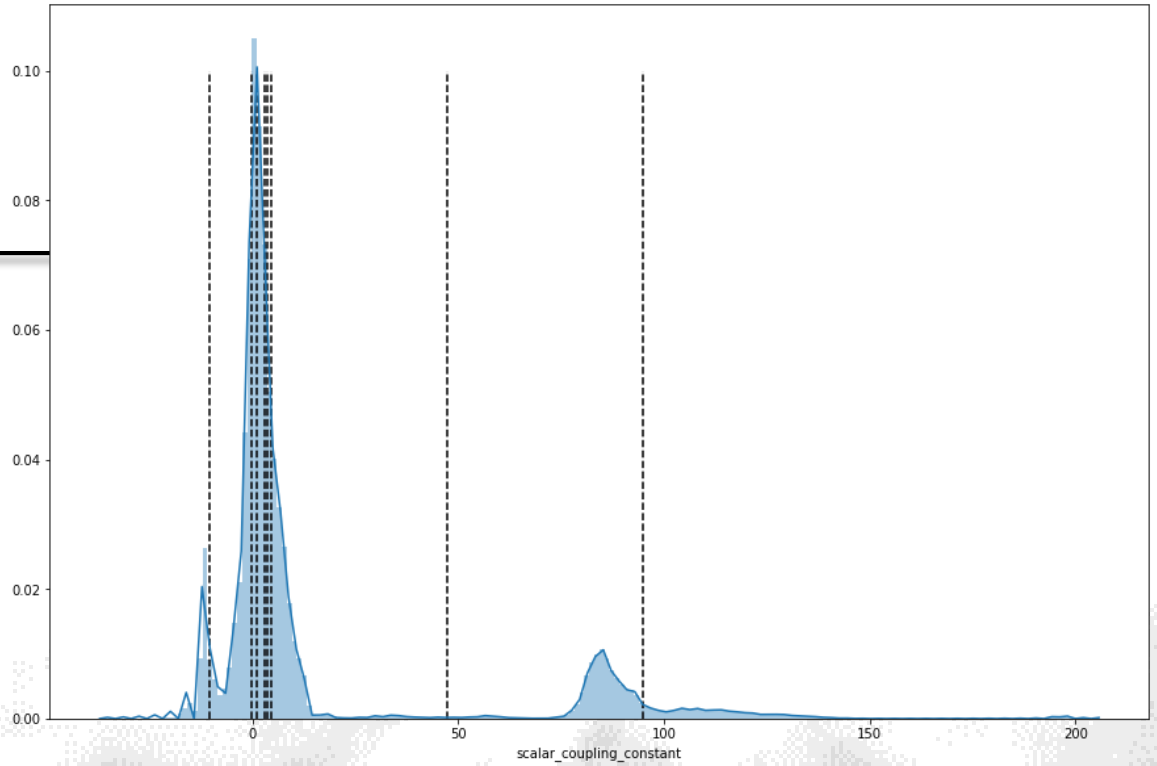
	molecule_name	atom_index	atom	x	y	z
0	dsgdb9nsd_000001	0	C	-0.012698	1.085804	0.008001
1	dsgdb9nsd_000001	1	H	0.002150	-0.006031	0.001976
2	dsgdb9nsd_000001	2	H	1.011731	1.463751	0.000277
3	dsgdb9nsd_000001	3	H	-0.540815	1.447527	-0.876644
4	dsgdb9nsd_000001	4	H	-0.523814	1.437933	0.906397

Data Exploring

Atom 3D structure



Coupling constant distribution



Atom coupling types and elements of atoms

H	1208387
C	831726
O	183187
N	132361
F	2996

3JHC	1510379
2JHC	1140674
1JHC	709416
3JHH	590611
2JHH	378036
3JHN	166415
2JHN	119253
1JHN	43363

Feature Engineering

Add more features by given parameters with domain knowledge:

- Relative distance between atoms
- Radius from the mass point
- Potential energy
- Number of atoms in molecule
- Number of neutrons and protons in atom

Add hidden information

- The coupling type between oxygen/fluorine and other atoms.
- The influence of oxygen/fluorine

Add graph influence(with Spark)

- The influence strength of different coupling types\elements for each atom in molecule

Data modelling – Deep learning

Schema: Build a neuron network model, set and tune the layers, activation function, estimators and loss to find the best performance

Requirement: the input data must be numeric data

```
def build_model():
    model = models.Sequential()
    model.add(layers.Dense(64, activation='relu',
                           input_shape=(train_X.shape[1],)))
    model.add(layers.Dense(64, activation='relu'))
    model.add(layers.Dense(96, activation='relu'))
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(1))
    model.compile(optimizer='rmsprop', loss='mse', metrics=['mae'])
    return model
```

```
## best epochs =185 by now
num_epochs = 185
model = build_model()
history = model.fit(train_X, train_y,
                    validation_data=(val_X, val_y),
                    epochs=num_epochs, batch_size=500, verbose=0)
mae_history = history.history['val_mean_absolute_error']
```

Activation function: 'RELU', 'ELU', 'PRELU', 'CRELU'

Optimizer: 'rmsprop', 'sigmoid', 'softmax'

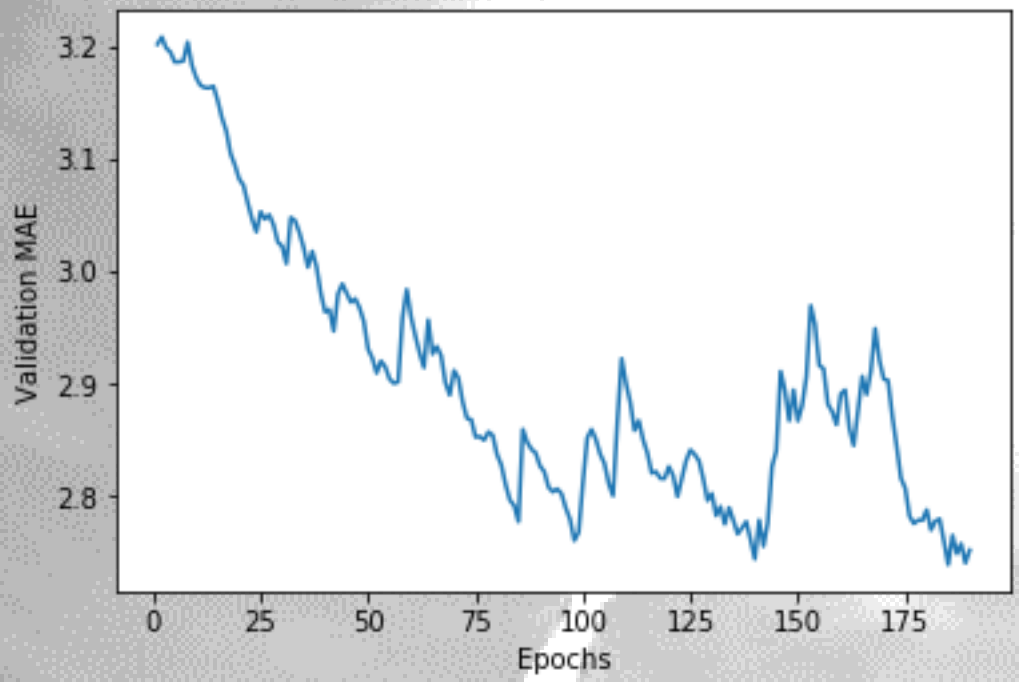
Loss: mse, ε-loss, hinge loss

Epoch: how many round to modify the weight

Batch size: train size for each modification

Prediction & Interpretation

K-fold validation



Choose number of epochs: 185

Result score: 0.744

Ranking on Kaggle: 1968

Thoughts

Spark with parallel:

I can't run spark graphframe with whole file. The Pregel thing will quickly exceed the memory. I need more machines to get all this work.

Spark limitation:

The pregel can only create one field at a time on atoms. It can't update the exist parameter and can't modify the relationships. So for my problem this is not meaningful to use Spark.

Efficiency:

Feeding such a large dataset to the model takes lots of time. So I tried to run it on AWS but AWS doesn't have the build-in image which saves the running result. I can't get a GPU permission either.

Summary:

The small graph problem is really a deep field to explore. It's especially inspiring on the feature engineering part, including building hidden effects.

Further to explore:

I'll use Neo4j to build the graph database and try to query the results. It allows fast queries for updating data.

I'll implement the code on the Google Cloud which has keras.