MASTER CODE - Material Mathematics Data Preprocessing

December 5, 2022

1 Material Mathematics (Group 5) Data Proposal Preprocessing

2 import packages

[1]:
```
! pip install keras_tuner
```

```
Requirement already satisfied: keras_tuner in /opt/conda/lib/python3.9/site-
packages (1.1.3)
Requirement already satisfied: tensorboard in /opt/conda/lib/python3.9/site-
packages (from keras_tuner) (2.4.1)
Requirement already satisfied: kt-legacy in /opt/conda/lib/python3.9/site-
packages (from keras_tuner) (1.0.4)
Requirement already satisfied: ipython in /opt/conda/lib/python3.9/site-packages
(from keras_tuner) (7.29.0)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-
packages (from keras_tuner) (21.2)
Requirement already satisfied: numpy in /opt/conda/lib/python3.9/site-packages
(from keras_tuner) (1.21.4)
Requirement already satisfied: requests in /opt/conda/lib/python3.9/site-
packages (from keras_tuner) (2.26.0)
Requirement already satisfied: decorator in /opt/conda/lib/python3.9/site-
packages (from ipython->keras_tuner) (5.1.0)
Requirement already satisfied: setuptools>=18.5 in
/opt/conda/lib/python3.9/site-packages (from ipython->keras_tuner) (58.5.3)
Requirement already satisfied: backcall in /opt/conda/lib/python3.9/site-
packages (from ipython->keras_tuner) (0.2.0)
Requirement already satisfied: pickleshare in /opt/conda/lib/python3.9/site-
packages (from ipython->keras_tuner) (0.7.5)
Requirement already satisfied: matplotlib-inline in
/opt/conda/lib/python3.9/site-packages (from ipython->keras_tuner) (0.1.3)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.9/site-
packages (from ipython->keras_tuner) (4.8.0)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.9/site-
packages (from ipython->keras_tuner) (0.18.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from ipython->keras_tuner) (3.0.22)
Requirement already satisfied: pygments in /opt/conda/lib/python3.9/site-
```

```
packages (from ipython->keras_tuner) (2.10.0)
Requirement already satisfied: traitlets>=4.2 in /opt/conda/lib/python3.9/site-
packages (from ipython->keras_tuner) (5.1.1)
Requirement already satisfied: pyparsing<3,>=2.0.2 in
/opt/conda/lib/python3.9/site-packages (from packaging->keras_tuner) (2.4.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.9/site-packages (from requests->keras_tuner) (1.26.7)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.9/site-packages (from requests->keras_tuner) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/site-
packages (from requests->keras_tuner) (3.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from requests->keras_tuner) (2.0.0)
Requirement already satisfied: grpcio>=1.24.3 in /opt/conda/lib/python3.9/site-
packages (from tensorboard->keras_tuner) (1.37.1)
Requirement already satisfied: google-auth<2,>=1.6.3 in
/opt/conda/lib/python3.9/site-packages (from tensorboard->keras_tuner) (1.35.0)
Requirement already satisfied: werkzeug>=0.11.15 in
/opt/conda/lib/python3.9/site-packages (from tensorboard->keras_tuner) (2.0.1)
Requirement already satisfied: six>=1.10.0 in /opt/conda/lib/python3.9/site-
packages (from tensorboard->keras_tuner) (1.15.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
/opt/conda/lib/python3.9/site-packages (from tensorboard->keras_tuner) (1.8.0)
Requirement already satisfied: protobuf>=3.6.0 in /opt/conda/lib/python3.9/site-
packages (from tensorboard->keras_tuner) (3.15.8)
Requirement already satisfied: wheel>=0.26 in /opt/conda/lib/python3.9/site-
packages (from tensorboard->keras_tuner) (0.37.0)
Requirement already satisfied: absl-py>=0.4 in /opt/conda/lib/python3.9/site-
packages (from tensorboard->keras_tuner) (0.15.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
/opt/conda/lib/python3.9/site-packages (from tensorboard->keras_tuner) (0.4.6)
Requirement already satisfied: markdown>=2.6.8 in /opt/conda/lib/python3.9/site-
packages (from tensorboard->keras_tuner) (3.3.4)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from google-
auth<2,>=1.6.3->tensorboard->keras_tuner) (4.2.4)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/lib/python3.9/site-packages (from google-
auth<2,>=1.6.3->tensorboard->keras_tuner) (0.2.7)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.9/site-
packages (from google-auth<2,>=1.6.3->tensorboard->keras_tuner) (4.7.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/lib/python3.9/site-packages (from google-auth-
oauthlib<0.5,>=0.4.1->tensorboard->keras_tuner) (1.3.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/opt/conda/lib/python3.9/site-packages (from jedi>=0.16->ipython->keras_tuner)
(0.8.2)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.9/site-
```

```
packages (from pexpect>4.3->ipython->keras_tuner) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.9/site-packages
(from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython->keras_tuner)
(0.2.5)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in
/opt/conda/lib/python3.9/site-packages (from pyasn1-modules>=0.2.1->google-
auth<2,>=1.6.3->tensorboard->keras_tuner) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in /opt/conda/lib/python3.9/site-
packages (from requests-oauthlib>=0.7.0->google-auth-
oauthlib<0.5,>=0.4.1->tensorboard->keras_tuner) (3.1.1)
```

[2]:
```
import numpy as np
from numpy import linalg as LA

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import Input
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.wrappers.scikit_learn import KerasRegressor

import keras_tuner as kt

import pickle as pkl
```

2.1 Hypothesis

In this proposal, we plan to find the correlation between each parameter and the solubility. We divide the parameters into two groups, trying to decide which parameter in each group has affected the solubility the most. One has a positive correlation with the molecules' solubility, and the other one has a negative correlation with the molecules' solubility.

In the positive correlation group, we guess that Number of H acceptors (NumHAcceptors) has the largest weight. Based on our current knowledge, because there is an equivalent partial negative charge on the atom bonded to hydrogen, this atom can accept H-bonds from water. Therefore, it

causes the compound's solubility to be better than compounds without H-bonds. Hydrogen bond strengths range from 4 kJ to 50 kJ per mole of hydrogen bonds. On the contrary, van der Waals interaction is a relatively weak force ranging from 0.5 to 1 kcal/mol.

In the negative correlation group, we hypothesize that the heavy atom count weighted the solubility the most. Since organic compounds are usually composed of C, O, and H, the higher molecular weight and more C make the compound more insoluble in water if H is not considered.

2.2 Data Import and Overview

In this section, we take a cursory look at the given dataset. We aim to understand each column and identify data relevant to our analysis.

[3]:
```
df = pd.read_csv("curated-solubility-dataset.csv")
```

[4]:
```
df.head(10)
```

[4]:

	ID	Name	InChI	InChIKey
0	A-3	N,N,N-trimethyloctadecan-1-aminium bromide	InChI=1S/C21H46N.BrH/c1-5-6-7-8-9-10-11-12-13-…	SZEMGTQCPRNXEG-UHFFFAOYSA-M
1	A-4	Benzo[cd]indol-2(1H)-one	InChI=1S/C11H7NO/c13-11-8-5-1-3-7-4-2-6-9(12-1…	GPYLCFQEKPUWLD-UHFFFAOYSA-N
2	A-5	4-chlorobenzaldehyde	InChI=1S/C7H5ClO/c8-7-3-1-6(5-9)2-4-7/h1-5H	AVPYQKSLYISFPO-UHFFFAOYSA-N
3	A-8	zinc bis[2-hydroxy-3,5-bis(1-phenylethyl)benzo…	InChI=1S/2C23H22O3.Zn/c2*1-15(17-9-5-3-6-10-17…	XTUPUYCJWKHGSW-UHFFFAOYSA-L
4	A-9	4-({4-[bis(oxiran-2-ylmethyl)amino]phenyl}meth…	InChI=1S/C25H30N2O4/c1-5-20(26(10-22-14-28-22)…	FAUAZXVRLVIARB-UHFFFAOYSA-N
5	A-10	vinyltoluene	InChI=1S/C9H10/c1-3-9-6-4-5-8(2)7-9/h3-7H,1H2,2H3	JZHGRUMIRATHIU-UHFFFAOYSA-N
6	A-11	3-(3-ethylcyclopentyl)propanoic acid	InChI=1S/C10H18O2/c1-2-8-3-4-9(7-8)5-6-10(11)1…		
7	A-12	11,16,17,21-tetrahydroxypregna-1,4-diene-3,20-…	InChI=1S/C21H28O6/c1-19-6-5-12(23)7-11(19)3-4-…		
8	A-14	bis(4-fluorophenyl)methanone	InChI=1S/C13H8F2O/c14-11-5-1-9(2-6-11)13(16)10…		
9	A-15	1-[2-(benzoyloxy)propoxy]propan-2-yl benzoate …	InChI=1S/C20H22O5/c21-19(17-9-3-1-4-10-17)24-1…		

```
6  WVRFSLWCFASCIS-UHFFFAOYSA-N
7  SEKYBDYVXDAYPY-UHFFFAOYSA-N
8  LSQARZALBDFYQZ-UHFFFAOYSA-N
9  BYQDGAVOOHIJQS-UHFFFAOYSA-N
```

	SMILES	Solubility	SD	\
0	[Br-].CCCCCCCCCCCCCCCCCC[N+](C)(C)C	-3.616127	0.000000	
1	O=C1Nc2cccc3cccc1c23	-3.254767	0.000000	
2	Clc1ccc(C=O)cc1	-2.177078	0.000000	
3	[Zn++].CC(c1ccccc1)c2cc(C(C)c3ccccc3)c(O)c(c2)…	-3.924409	0.000000	
4	C1OC1CN(CC2CO2)c3ccc(Cc4ccc(cc4)N(CC5CO5)CC6CO…	-4.662065	0.000000	
5	Cc1cccc(C=C)c1	-3.123150	0.000000	
6	CCC1CCC(CCC(O)=O)C1	-3.286116	0.000000	
7	CC12CC(O)C3C(CCC4=CC(=O)C=CC34C)C1CC(O)C2(O)C(…	-2.664549	0.000000	
8	Fc1ccc(cc1)C(=O)c2ccc(F)cc2	-4.396652	0.431513	
9	O=C(OCCCOCCCOC(=O)c1ccccc1)c2ccccc2	-4.595503	0.118551	

	Ocurrences	Group	MolWt	…	NumRotatableBonds	NumValenceElectrons	\
0	1	G1	392.510	…	17.0	142.0	
1	1	G1	169.183	…	0.0	62.0	
2	1	G1	140.569	…	1.0	46.0	
3	1	G1	756.226	…	10.0	264.0	
4	1	G1	422.525	…	12.0	164.0	
5	1	G1	118.179	…	1.0	46.0	
6	1	G1	170.252	…	4.0	70.0	
7	1	G1	376.449	…	2.0	148.0	
8	2	G3	218.202	…	2.0	80.0	
9	2	G3	342.391	…	10.0	132.0	

	NumAromaticRings	NumSaturatedRings	NumAliphaticRings	RingCount	TPSA	\
0	0.0	0.0	0.0	0.0	0.00	
1	2.0	0.0	1.0	3.0	29.10	
2	1.0	0.0	0.0	1.0	17.07	
3	6.0	0.0	0.0	6.0	120.72	
4	2.0	4.0	4.0	6.0	56.60	
5	1.0	0.0	0.0	1.0	0.00	
6	0.0	1.0	1.0	1.0	37.30	
7	0.0	3.0	4.0	4.0	115.06	
8	2.0	0.0	0.0	2.0	17.07	
9	2.0	0.0	0.0	2.0	61.83	

	LabuteASA	BalabanJ	BertzCT
0	158.520601	0.000000e+00	210.377334
1	75.183563	2.582996e+00	511.229248
2	58.261134	3.009782e+00	202.661065
3	323.755434	2.322963e-07	1964.648666
4	183.183268	1.084427e+00	769.899934

```
5   55.836626  3.070761e+00   211.033225
6   73.973655  2.145839e+00   153.917569
7  158.135542  1.776978e+00   755.770792
8   91.346032  2.315628e+00   452.960733
9  147.071714  1.447050e+00   582.150793

[10 rows x 26 columns]
```

[5]:
```
df.columns
```

[5]:
```
Index(['ID', 'Name', 'InChI', 'InChIKey', 'SMILES', 'Solubility', 'SD',
       'Ocurrences', 'Group', 'MolWt', 'MolLogP', 'MolMR', 'HeavyAtomCount',
       'NumHAcceptors', 'NumHDonors', 'NumHeteroatoms', 'NumRotatableBonds',
       'NumValenceElectrons', 'NumAromaticRings', 'NumSaturatedRings',
       'NumAliphaticRings', 'RingCount', 'TPSA', 'LabuteASA', 'BalabanJ',
       'BertzCT'],
      dtype='object')
```

[6]:
```
df["Group"]
```

[6]:
```
0       G1
1       G1
2       G1
3       G1
4       G1
       ..
9977    G1
9978    G1
9979    G5
9980    G1
9981    G5
Name: Group, Length: 9982, dtype: object
```

[7]:
```
df.keys()
```

[7]:
```
Index(['ID', 'Name', 'InChI', 'InChIKey', 'SMILES', 'Solubility', 'SD',
       'Ocurrences', 'Group', 'MolWt', 'MolLogP', 'MolMR', 'HeavyAtomCount',
       'NumHAcceptors', 'NumHDonors', 'NumHeteroatoms', 'NumRotatableBonds',
       'NumValenceElectrons', 'NumAromaticRings', 'NumSaturatedRings',
       'NumAliphaticRings', 'RingCount', 'TPSA', 'LabuteASA', 'BalabanJ',
       'BertzCT'],
      dtype='object')
```

[8]:
```
df.dtypes
```

[8]:
```
ID                    object
Name                  object
```

```
InChI                  object
InChIKey               object
SMILES                 object
Solubility            float64
SD                    float64
Ocurrences              int64
Group                  object
MolWt                 float64
MolLogP               float64
MolMR                 float64
HeavyAtomCount        float64
NumHAcceptors         float64
NumHDonors            float64
NumHeteroatoms        float64
NumRotatableBonds     float64
NumValenceElectrons   float64
NumAromaticRings      float64
NumSaturatedRings     float64
NumAliphaticRings     float64
RingCount             float64
TPSA                  float64
LabuteASA             float64
BalabanJ              float64
BertzCT               float64
dtype: object
```

As the data type of each column listed above, the first 9 columns are the names, various identifiers of the compounds, and the original dataset before merging. They are irrelevant to our analysis because they are not traits or properties that describe or uniquely identify the compound's solubility. Therefore, we exclude them from our data analysis in the latter sections.

[9]:
```
df.isna().sum()
```

[9]:
```
ID                    0
Name                  0
InChI                 0
InChIKey              0
SMILES                0
Solubility            0
SD                    0
Ocurrences            0
Group                 0
MolWt                 0
MolLogP               0
MolMR                 0
HeavyAtomCount        0
NumHAcceptors         0
```

```
NumHDonors              0
NumHeteroatoms          0
NumRotatableBonds       0
NumValenceElectrons     0
NumAromaticRings        0
NumSaturatedRings       0
NumAliphaticRings       0
RingCount               0
TPSA                    0
LabuteASA               0
BalabanJ                0
BertzCT                 0
dtype: int64
```

The given data has already been filtered, and no NaN existed. We don't need to process the raw data further.

2.3 Training-Testing Split

We separate the training and testing data to avoid potential bias. Additionally, we save the split data so that group members can work on separate models using the same dataset. Therefore, we can compare different models fairly in the end.

[10]:
```python
class Archive:
    def __init__(self, file_path=None):
        if (file_path is None):
            self.file_path = ""
        else:
            self.file_path = file_path

    def save_data(self, data, file_name):
        with open("{}{}".format(self.file_path, file_name), 'wb') as f:
            pkl.dump(data, f)

    def load_data(self, file_name):
        with open("{}{}".format(self.file_path, file_name), 'rb') as f:
            return pkl.load(f)

data_directory = Archive()
```

The relevent labels of input and output are the following:

[11]:
```python
# Labels of input and output matrices
X_label = [
    'MolWt', 'MolLogP', 'MolMR', 'HeavyAtomCount',
    'NumHAcceptors', 'NumHDonors', 'NumHeteroatoms', 'NumRotatableBonds',
    'NumValenceElectrons', 'NumAromaticRings', 'NumSaturatedRings',
    'NumAliphaticRings', 'RingCount', 'TPSA', 'LabuteASA', 'BalabanJ',
```

```
    'BertzCT'
]
y_label = ['Solubility']
```

[12]:
```
# Shape the input and output matrices in correct format
X_nparr = df[X_label].values
y_nparr = df[y_label].values.reshape(-1)
```

We save our split data into `Solubility Data Splitted.pkl` file.

The following cell will save the dataset for the first time, but it should remain commented afterwards because we don't want to overwrite them.

[13]:
```
# save check point

# X_train, X_test, y_train, y_test = train_test_split(X_nparr, y_nparr,␣
 ↪test_size=0.1)
# X_train, X_validation, y_train, y_validation = train_test_split(X_train,␣
 ↪y_train, test_size=0.2)
# data_directory.save_data([X_train, y_train, X_validation, y_validation,␣
 ↪X_test, y_test], "Solubility Data Splitted.pkl")
```

We load the data into respective variables.

[14]:
```
# load check point

# only import training and validation data
X_train, y_train, X_validation, y_validation, _, _ = data_directory.
 ↪load_data("Solubility Data Splitted.pkl")
```

We normalized the input to avoid bias induced by the different magnitudes of data, which is essential for neural networks and clustering algorithms. We don't have to normalize the output because it only has one dimension.

[15]:
```
# Standardize the input and output
X_scaler = StandardScaler().fit(np.concatenate([X_train, X_validation], axis=0))

# training, validation combined
X_nparr_normalize = X_scaler.transform(np.concatenate([X_train, X_validation]))
y_nparr = np.concatenate([y_train, y_validation], axis=0)

# training data
X_train_normalized = X_scaler.transform(X_train)

# validation data
X_validation_normalized = X_scaler.transform(X_validation)
```

Model Input Data: - `X_train`, `X_validation`: input, not normalized - `X_train_normalized`, `X_validation_normalized`: input, normalized version of the data above

Model Output Data: - `y_train`, `y_validation`: output, not normalized

2.4 Linear Regression Model

In this section, we will use linear regression to show the significance of each parameter when predicting solubility.

[16]:
```
model = LinearRegression().fit(X_nparr_normalize, y_nparr)
```

[17]:
```
model.coef_
```

[17]:
```
array([-1.08590361, -1.62080856,  0.8561797 , -6.22212344,  0.27107273,
        0.21092471, -0.5168519 ,  0.23234583,  5.34339449, -0.48439856,
        0.10317919, -0.09553739, -0.44851903, -0.22987086, -0.7816085 ,
       -0.03861738,  2.55568672])
```

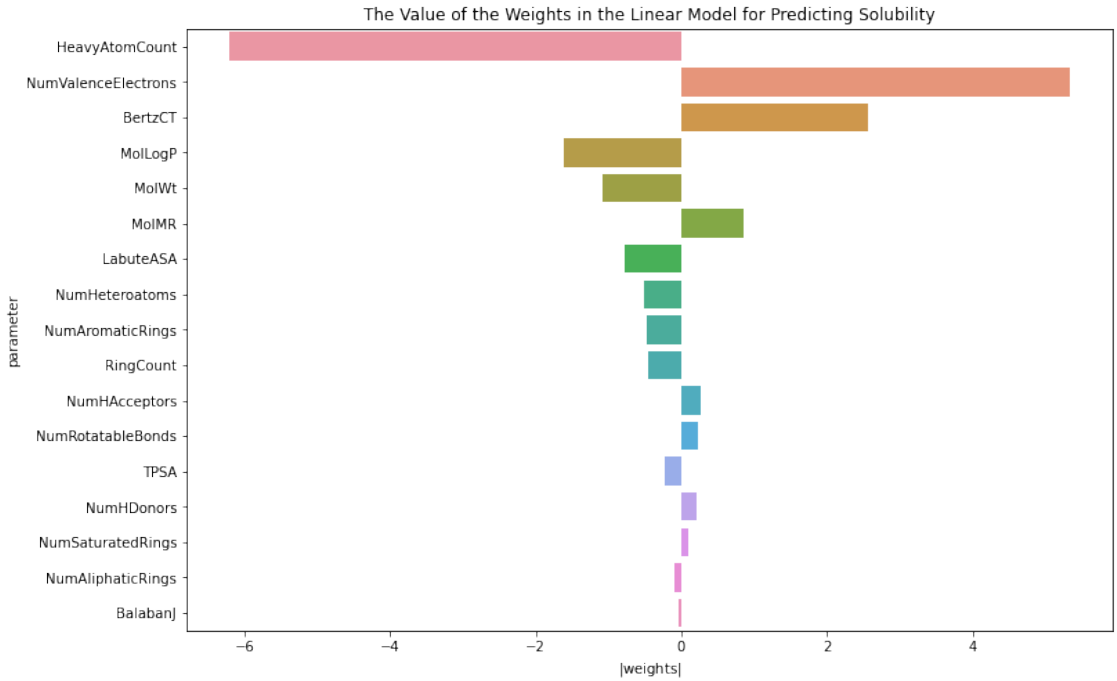
2.4.1 Linear Model Weights

[18]:
```
weights = list(zip(X_label, model.coef_))
weights.sort(key=lambda x: np.abs(x[1]), reverse=True)
display(pd.DataFrame(weights))
```

	0	1
0	HeavyAtomCount	-6.222123
1	NumValenceElectrons	5.343394
2	BertzCT	2.555687
3	MolLogP	-1.620809
4	MolWt	-1.085904
5	MolMR	0.856180
6	LabuteASA	-0.781608
7	NumHeteroatoms	-0.516852
8	NumAromaticRings	-0.484399
9	RingCount	-0.448519
10	NumHAcceptors	0.271073
11	NumRotatableBonds	0.232346
12	TPSA	-0.229871
13	NumHDonors	0.210925
14	NumSaturatedRings	0.103179
15	NumAliphaticRings	-0.095537
16	BalabanJ	-0.038617

[19]:
```
df_sns = pd.DataFrame(
    [{"parameter": weight[0], "linear weight": weight[1]} for weight in weights]
)

fig, ax = plt.subplots(1, 1, figsize=(12, 8))
sns.barplot(
    data=df_sns,
```

```
    x="linear weight",
    y="parameter",
    ax=ax
)
ax.set_title("The Value of the Weights in the Linear Model for Predicting␣
 ↪Solubility")
ax.set_xlabel("$|$weights$|$")
plt.show()
```



From the figure above, we have more parameters that are negatively correlated to solubility than others. Furthermore, the absolute value of the negatively correlated parameter are tend to be larger.

[20]:
```
for i in np.arange(5):
    print(weights[i])
```

```
('HeavyAtomCount', -6.222123444135655)
('NumValenceElectrons', 5.343394494009954)
('BertzCT', 2.5556867170521707)
('MolLogP', -1.6208085606823759)
('MolWt', -1.0859036120957708)
```

The parameters above might be the first 5 most important parameter to predict solubility of compounds. We need to pay attention on them later in the project.

[21]:
```
# save model for testing
linear_regression_model = model
```

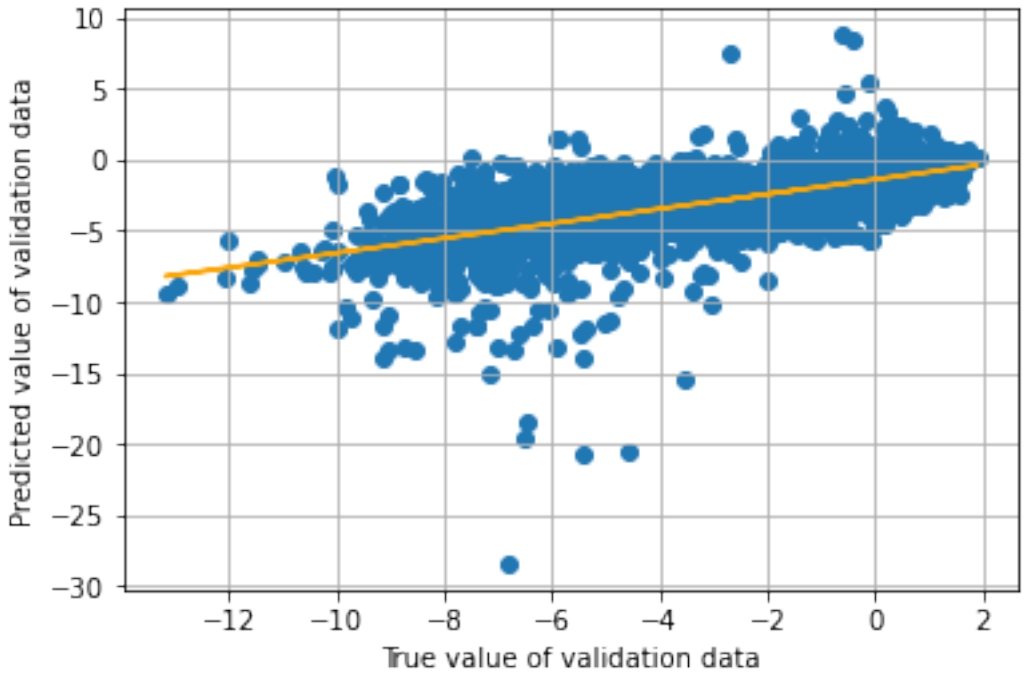
2.4.2 Variance (r^2) and Parity Plot

[22]:
```
print("Variance of the linear model is {0}".format(r2_score(y_nparr, model.
 ↪predict(X_nparr_normalize))))
```

```
Variance of the linear model is 0.517435867766471
```

Linear regression model is not a good selection to predict the solubility of given parameter because the variance is low (We expect that a good model will have at least 0.80 variance).

[23]:
```
y_nparr_predict = model.predict(X_nparr_normalize)

plt.scatter(y_nparr, y_nparr_predict)

# line of best fit (linear regression) on the parity plot.
coeff = np.polyfit(y_nparr, y_nparr_predict, 1)
y_nparr_predict = np.polyval(coeff, y_nparr)
plt.plot(y_nparr, y_nparr_predict, c = 'orange')  # parity_plot (true_data,␣
 ↪predicted_data)

plt.xlabel("True value of validation data")
plt.ylabel("Predicted value of validation data")
plt.grid()
plt.show()
```



[24]:
```
print("The slope of the fitted line is {0}.".format(np.polyfit(y_nparr, model.
 ↪predict(X_nparr_normalize), 1)[0]))
```

```
The slope of the fitted line is 0.5174358677664681.
```

The slope in the perfectly fitted line is 1. Therefore, linear regression model is not powerful enough to predict solubility accurately.

3 Neural Networks

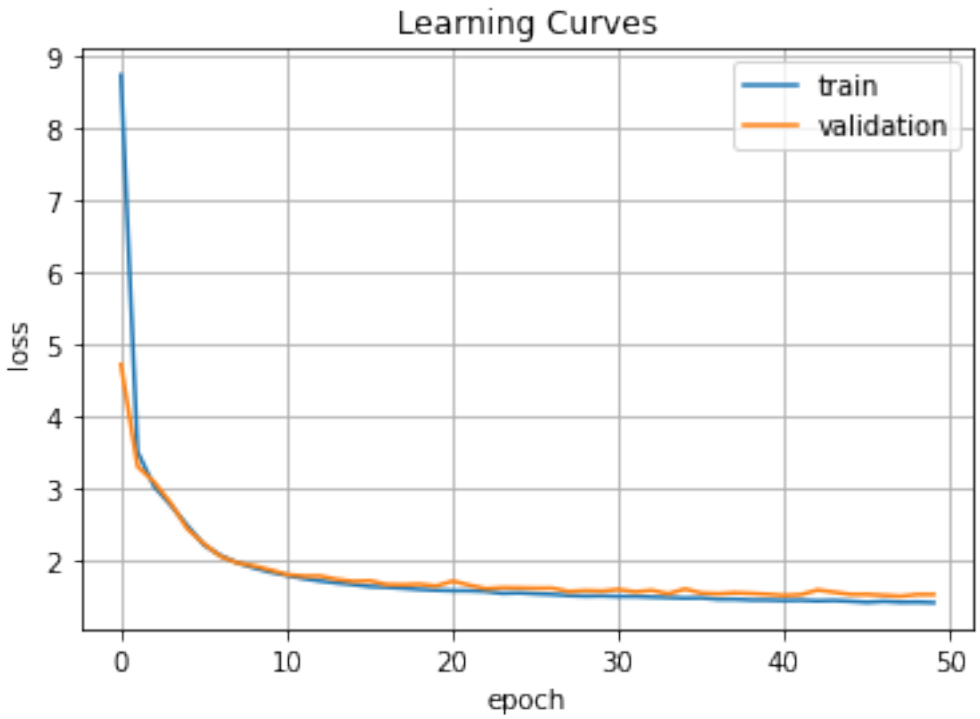
[25]:
```
#Define a neural network model, and wrap this model in a function
def nnmodel():
    model = Sequential()
    model.add(Dense(17, input_dim=X_train_normalized.shape[1],␣
 ↪kernel_initializer='normal', activation='relu'))
    model.add(Dense(34, kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
    model.compile(loss='mean_squared_error', optimizer='Adam')
    return model
```

[26]:
```
estimator = KerasRegressor(build_fn=nnmodel) #nnmodel1 is the neural network␣
 ↪model we defined above
history = estimator.fit(X_train_normalized, y_train, validation_split=0.3,␣
 ↪epochs=50, batch_size=34, verbose=0)
```

```
2022-12-04 11:35:36.282348: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-12-04 11:35:36.436338: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2022-12-04 11:35:36.437449: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency:
2199995000 Hz
```

[27]:
```
plt.plot(range(50),history.history['loss'])
plt.plot(range(50),history.history['val_loss'])
plt.title('Learning Curves')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper right')
plt.grid()
```

```
plt.show()
```



[28]:
```
print("final MSE for train is %.2f and for validation is %.2f" %
      (history.history['loss'][-1], history.history['val_loss'][-1]))
```

```
final MSE for train is 1.42 and for validation is 1.53
```

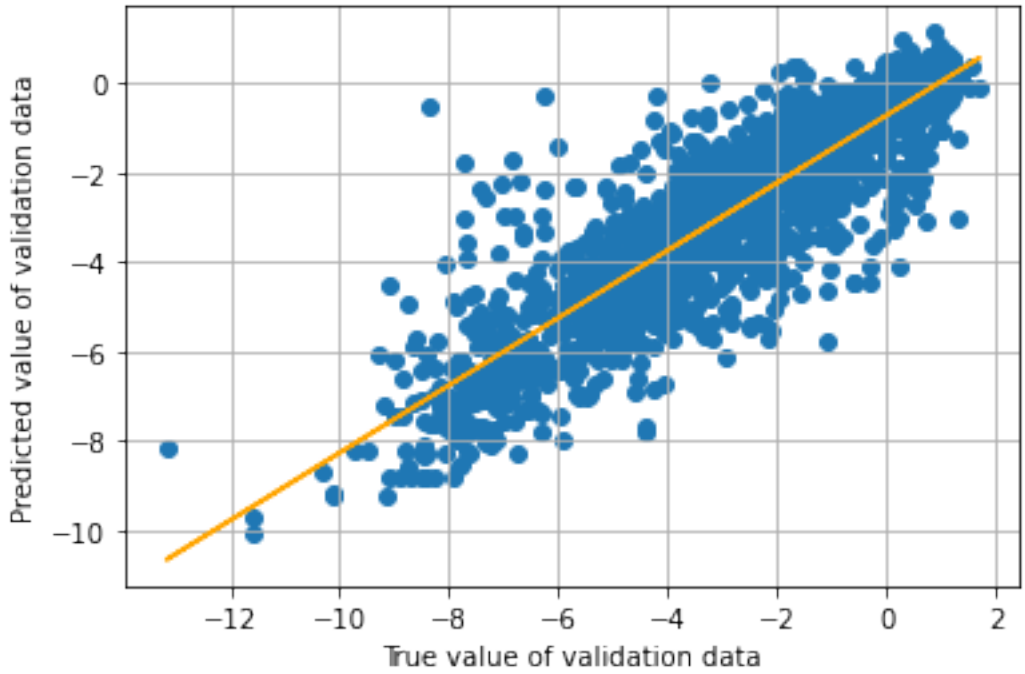
[29]:
```
test_loss = estimator.model.evaluate(X_validation_normalized, y_validation)
print("test set mse is %.2f" % test_loss)
```

```
57/57 [==============================] - 0s 724us/step - loss: 1.5040
test set mse is 1.50
```

[30]:
```
y_validation_predict = estimator.predict(X_validation_normalized)
```

[31]:
```
plt.scatter(y_validation, y_validation_predict)

# line of best fit (linear regression) on the parity plot.
coeff = np.polyfit(y_validation, y_validation_predict, 1)
y_validation_predict_ = np.polyval(coeff, y_validation)
plt.plot(y_validation, y_validation_predict_, c = 'orange')  # parity_plot␣
↪(true_data, predicted_data)

plt.xlabel("True value of validation data")
```

```
plt.ylabel("Predicted value of validation data")
plt.grid()
plt.show()
```



[32]:
```
print("Variance of the neural network model is {0}".
 ↪format(r2_score(y_validation, y_validation_predict)))
```

```
Variance of the neural network model is 0.7493783074034881
```

[33]:
```
# save model for testing
neural_network_model = estimator.model
```

4 Principal Component Analysis and Neural Network

4.1 PCA

Principal component Analysis (PCA) is a method of unsupervised learning for feature extraction so that we can extract the features by importance. PCA is also used for dimensionality reduction to reduce high dimension dataset into lower dimension by grouping the features in the unsupervised approach. By grouping some of the original features, some information will be lost so we need to decide how many of the "new" features that we want to have in the reduced dataset.

To do that we can plot the cumulative variance as the number of features.

4.1.1 [!!!MODIFICATION NEEDED] Mathematical Perspective (SVD & Covariance)

[34]:
```
max_n_pcs = X_nparr_normalize.shape[1]
```

We will apply PCA on all the dimension of the data and we will plot the explained variance ratio to see what is the minimum components we can use so that we can preserve the data up to 90% with the reduced data dimension.

[35]:
```
pca_17 = PCA(n_components=max_n_pcs)
pca_17.fit(X_nparr_normalize)
```

[35]:
```
PCA(n_components=17)
```

[36]:
```
most_important_features_index = [np.abs(pca_17.components_[i]).argmax() for i␣
↪in range(max_n_pcs)]
most_important_features_index
```

[36]:
```
[3, 1, 11, 9, 15, 5, 7, 10, 4, 13, 7, 16, 16, 14, 2, 3, 12]
```

4.1.2 The order of features from most to least important

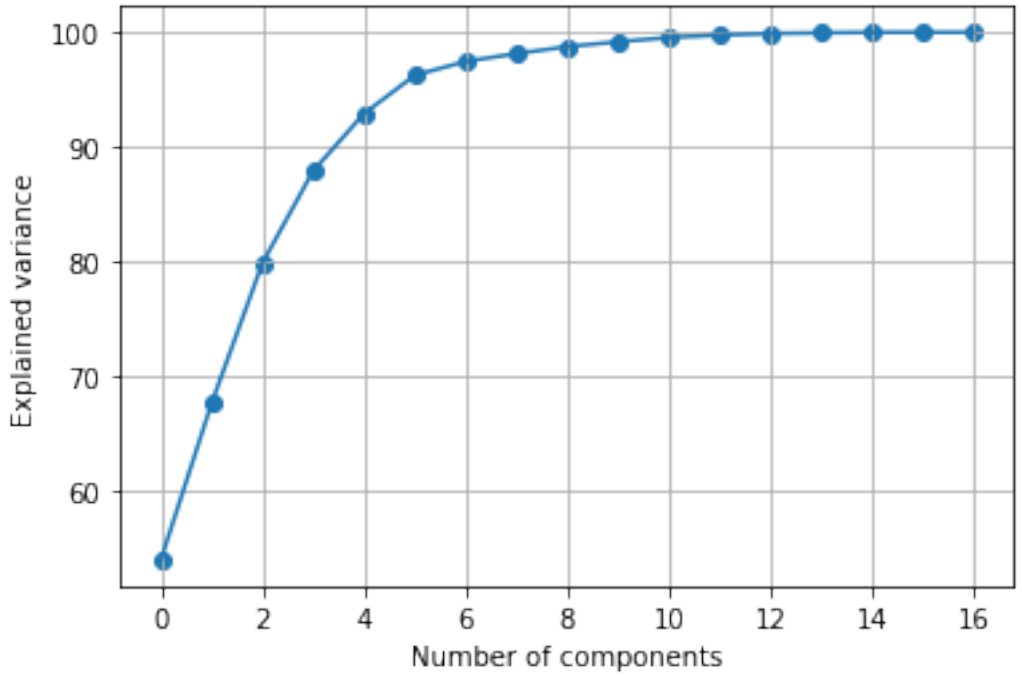
[37]:
```
most_important_features = [X_label[i] for i in most_important_features_index]
most_important_features
```

[37]:
```
['HeavyAtomCount',
 'MolLogP',
 'NumAliphaticRings',
 'NumAromaticRings',
 'BalabanJ',
 'NumHDonors',
 'NumRotatableBonds',
 'NumSaturatedRings',
 'NumHAcceptors',
 'TPSA',
 'NumRotatableBonds',
 'BertzCT',
 'BertzCT',
 'LabuteASA',
 'MolMR',
 'HeavyAtomCount',
 'RingCount']
```

[38]:
```
plt.grid()
plt.scatter(range(0,17),np.cumsum(pca_17.explained_variance_ratio_ * 100))
plt.plot(range(0,17),np.cumsum(pca_17.explained_variance_ratio_ * 100))
plt.xlabel('Number of components')
```

```
plt.ylabel('Explained variance')
```

[38]:
```
Text(0, 0.5, 'Explained variance')
```



We will use 5 components of PCA since it will give us at least 90% attributes of the original dataset.

[39]:
```
pca = PCA(n_components=5)
pca.fit(X_nparr_normalize)
```

[39]:
```
PCA(n_components=5)
```

[40]:
```
reduced_x_nparr_normal = pca.transform(X_nparr_normalize)
```

[41]:
```
reduced_x_train_normal = pca.transform(X_train_normalized)
```

[42]:
```
reduced_x_train_normal.shape
```

[42]:
```
(7186, 5)
```

[43]:
```
reduced_x_validation_normal = pca.transform(X_validation_normalized)
```

[44]:
```
# reduced_x_test = pca.transform(X_test_normalized)
```

4.2 NN

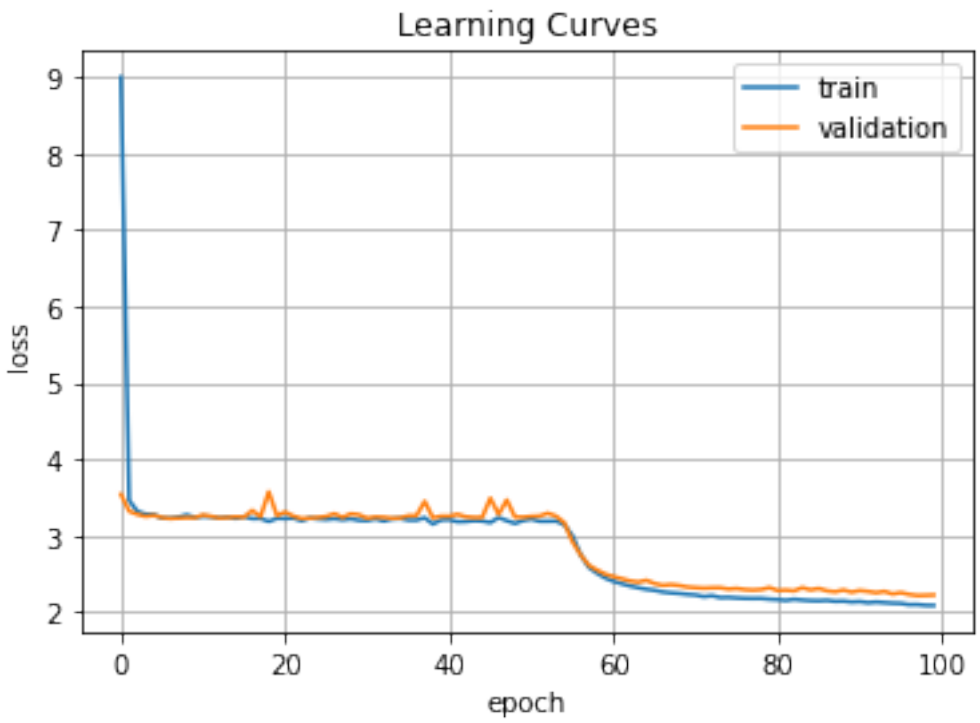
[45]:
```
#Define a neural network model, and wrap this model in a function
def nn_PCA_model():
    model = Sequential()
    model.add(Dense(16, input_dim=reduced_x_nparr_normal.shape[1],␣
 ↪kernel_initializer='normal', activation='relu'))
    model.add(Dense(7, kernel_initializer='normal', activation='relu'))
    #model.add(Dense(8, kernel_initializer='normal', activation='relu'))
    model.add(Dense(4,kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model
```

[46]:
```
estimator = KerasRegressor(build_fn=nn_PCA_model) #nnmodel1 is the neural␣
 ↪network model we defined above
history = estimator.fit(reduced_x_nparr_normal, y_nparr, validation_split=0.2,␣
 ↪epochs=100, batch_size=32, verbose=0)
```

[47]:
```
estimator.model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_3 (Dense)              (None, 16)                96
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 119
_________________________________________________________________
dense_5 (Dense)              (None, 4)                 32
_________________________________________________________________
dense_6 (Dense)              (None, 1)                 5
=================================================================
Total params: 252
Trainable params: 252
Non-trainable params: 0
_________________________________________________________________
```

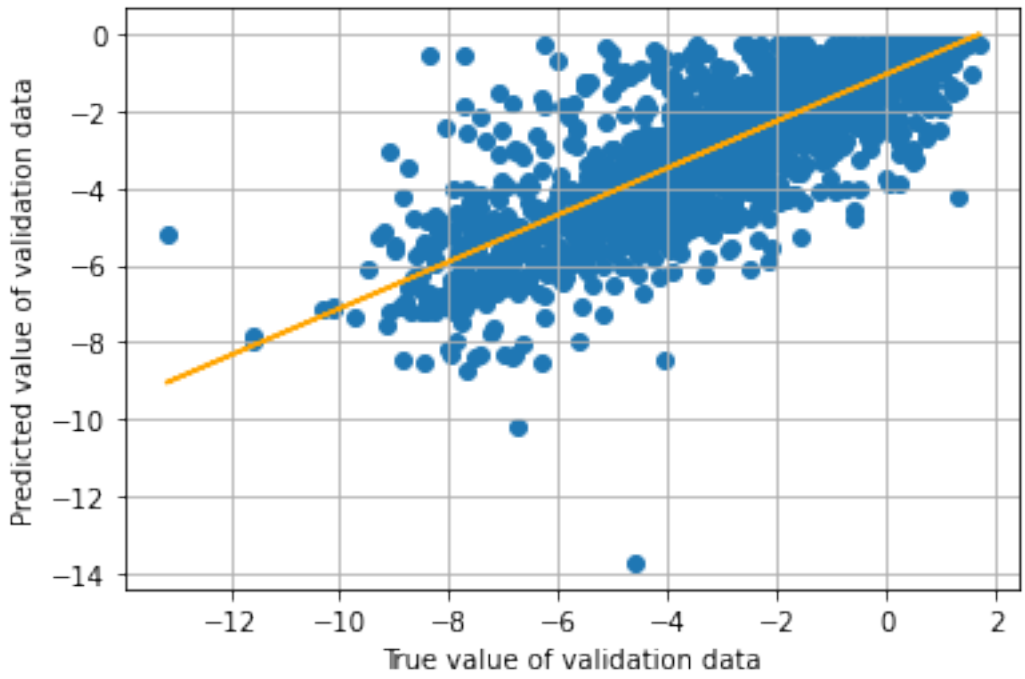
[48]:
```
plt.plot(range(100),history.history['loss'])
plt.plot(range(100),history.history['val_loss'])
plt.title('Learning Curves')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper right')
plt.grid()
plt.show()
```



[49]:
```
print("final MSE for train is %.2f and for validation is %.2f" %
      (history.history['loss'][-1], history.history['val_loss'][-1]))
```

```
final MSE for train is 2.09 and for validation is 2.23
```

[50]:
```
y_validation_predict3 = estimator.predict(reduced_x_validation_normal)
plt.scatter(y_validation, y_validation_predict3)

# line of best fit (linear regression) on the parity plot.
coeff = np.polyfit(y_validation, y_validation_predict3, 1)
y_validation_predict3_ = np.polyval(coeff, y_validation)
plt.plot(y_validation, y_validation_predict3_, c = 'orange')  # parity_plot␣
↪(true_data, predicted_data)

plt.xlabel("True value of validation data")
plt.ylabel("Predicted value of validation data")
plt.grid()
plt.show()
```



[51]:
```
print("Variance of the PCA+NN model is {0}".format(r2_score(y_validation,␣
↪y_validation_predict3)))
```

```
Variance of the PCA+NN model is 0.6291158613600037
```

4.2.1 Hyperparameter Tuning NN

[52]:
```
def regression_nn_builder(hp):

    '''
    This is a builder function to define and compile a neural network

    Parameters:
    '''
    a = np.random.randint(5,15,size=1)
    b = np.random.randint(1,10,size=1)
    model = Sequential()
    model.add(Dense(units=hp.Int("units_0", min_value=1, max_value=20, step=1),
                    input_dim=reduced_x_nparr_normal.shape[1],␣
↪kernel_initializer='normal', activation='relu'))
    model.add(Dense(units=hp.Int("units_1", min_value=1, max_value=20, step=1),␣
↪kernel_initializer='normal', activation='relu'))
    model.add(Dense(units=hp.Int("units_2", min_value=1, max_value=20,␣
↪step=1),kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
```

```
    hp_learning_rate = hp.Choice('learning_rate', values=[5e-3,3e-3, 1e-3,␣
↪5e-4,1e-4])

    model.compile(loss='mean_squared_error',
                  optimizer=keras.optimizers.
↪Adam(learning_rate=hp_learning_rate),metrics=[keras.metrics.
↪MeanSquaredError()])
    return model
```

[53]:
```
# set the build function to be binary_nn_builder
# set the default value of batch_size to be 32

tuner = kt.RandomSearch(
    hypermodel=regression_nn_builder,
    objective=kt.Objective("val_mean_squared_error", direction="min"),
    max_trials=40,
    executions_per_trial=3,
    overwrite=True,
    directory="tuning_hyperparameter1",
    project_name="Trial2",
)
```

[54]:
```
tuner.search_space_summary()
```

```
Search space summary
Default search space size: 4
units_0 (Int)
{'default': None, 'conditions': [], 'min_value': 1, 'max_value': 20, 'step': 1,
'sampling': None}
units_1 (Int)
{'default': None, 'conditions': [], 'min_value': 1, 'max_value': 20, 'step': 1,
'sampling': None}
units_2 (Int)
{'default': None, 'conditions': [], 'min_value': 1, 'max_value': 20, 'step': 1,
'sampling': None}
learning_rate (Choice)
{'default': 0.005, 'conditions': [], 'values': [0.005, 0.003, 0.001, 0.0005,
0.0001], 'ordered': True}
```

[55]:
```
tuner.search(reduced_x_train_normal, y_train, epochs=100,␣
↪validation_data=(reduced_x_validation_normal, y_validation))
```

```
Trial 40 Complete [00h 01m 50s]
val_mean_squared_error: 2.1011576652526855

Best val_mean_squared_error So Far: 1.8535271485646565
Total elapsed time: 01h 07m 10s
```

```
INFO:tensorflow:Oracle triggered exit
```

[56]:
```
# Get the top 2 models.
models = tuner.get_best_models(num_models=2)
best_model = models[0]
# Build the model.
# Needed for `Sequential` without specified `input_shape`.
best_model.build(input_shape=reduced_x_nparr_normal.shape[1])
best_model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 16)                96
_________________________________________________________________
dense_1 (Dense)              (None, 12)                204
_________________________________________________________________
dense_2 (Dense)              (None, 18)                234
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 19
=================================================================
Total params: 553
Trainable params: 553
Non-trainable params: 0
_________________________________________________________________
```

[59]:
```
best_model.evaluate(reduced_x_validation_normal, y_validation)
```

```
57/57 [==============================] - 0s 2ms/step - loss: 1.9149 -
mean_squared_error: 1.9149
```

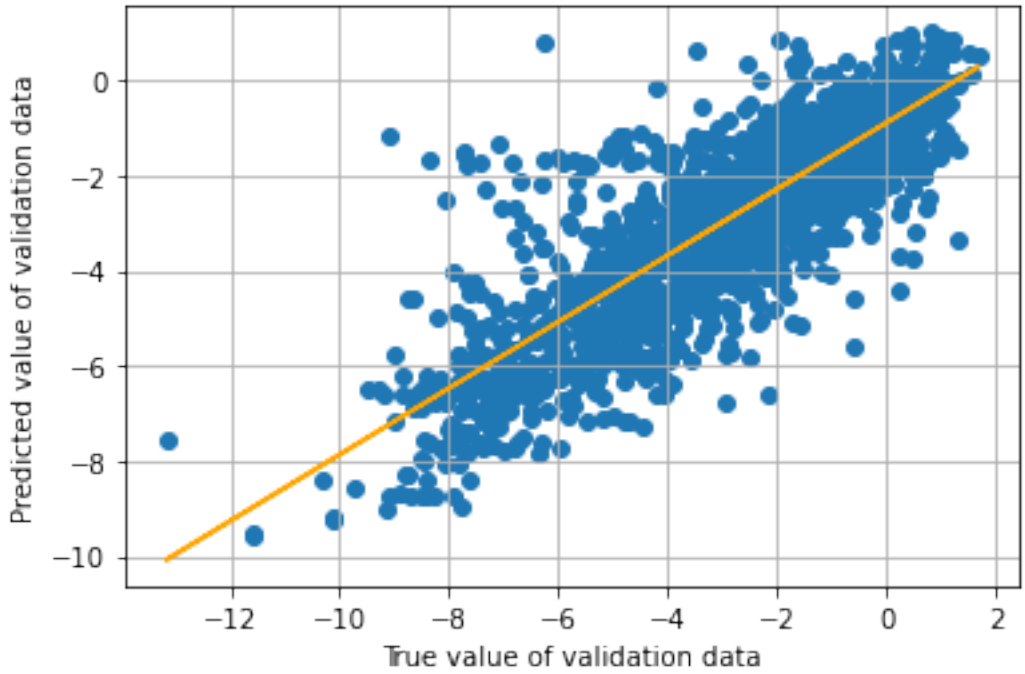
[59]:
```
[1.8041058778762817, 1.8041058778762817]
```

[60]:
```
mean_squared_error(y_validation, best_model.
 ↪predict(reduced_x_validation_normal))
```

[60]:
```
1.8041058459553296
```

[71]:
```
y_validation_predict3 = best_model.predict(reduced_x_validation_normal)
plt.scatter(y_validation, y_validation_predict3)

# line of best fit (linear regression) on the parity plot.
coeff = np.polyfit(y_validation, y_validation_predict3, 1)
y_validation_predict3_ = np.polyval(coeff, y_validation)
plt.plot(y_validation, y_validation_predict3_, c = 'orange')  # parity_plot␣
 ↪(true_data, predicted_data)
```

```
plt.xlabel("True value of validation data")
plt.ylabel("Predicted value of validation data")
plt.grid()
```



[72]:
```
print("Variance of the PCA+NN model (after fine tuning) is {0}".
 ↪format(r2_score(y_validation, y_validation_predict3)))
```

```
Variance of the PCA+NN model (after fine tuning) is 0.6993708414899424
```

[73]:
```
# save model for testing
PCA_NN_model = {}
PCA_NN_model["PCA"] = pca
PCA_NN_model["nn"] = best_model
```

5 K-Mean Cluster and Neural Network

K-mean Clustering (KMC) is an unsupervised machine-learning technique that groups similar data. Compared to PCA, which reduces the dimension of the data, grouping by K-mean Clustering reduces the mean variance of the dataset. Also, the neural network's complexity depends on the variance of datasets. In other words, if the dataset is simple, we can dramatically reduce the complexity of neural networks, resulting in models with fewer hidden layers and fewer perceptron.

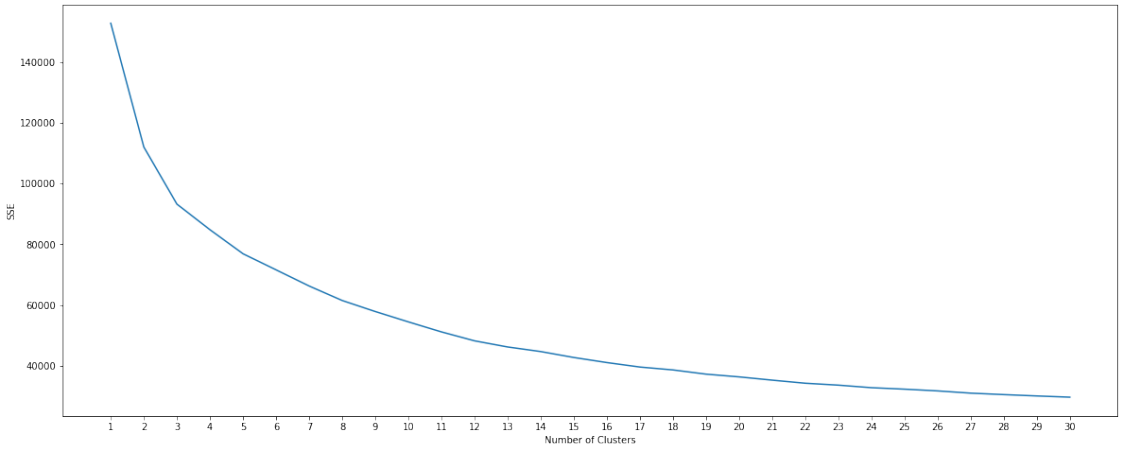
Thus, we hypothesize that running a neural network on each group would return better results because the variance of each group is significantly smaller than that of the entire dataset.

5.1 KMC

5.1.1 Finding Optimal Groups

We use Elbow Mehod to find the optimal group for this dataset.

[74]:
```
sse = []
x_plot = np.arange(30) + 1
for k in x_plot:
    KMC_Model = KMeans(
        n_clusters=k,
        n_init=10,
        init='k-means++',
        max_iter=300,
    ).fit(
        X_nparr_normalize
    )
    sse.append(KMC_Model.inertia_)

plt.figure(figsize=(20,8))
plt.plot(x_plot, sse)
plt.xticks(x_plot)
plt.xlabel("Number of Clusters")
plt.ylabel("SSE")
plt.show()
```



At $k = 5$, the sum of the squared error is small while the number of groups is not large.

5.1.2 Calculation

[75]:
```
# fit input by K-Mean clustering
num_clusters = 5
KMC_Model = KMeans(
    n_clusters=num_clusters,
    n_init=10,
    init='k-means++',
    max_iter=300,
).fit(
    X_nparr_normalize
)
```

[76]:
```
# calculate distance between each sample and its respective centroid.
dist = []
for centroid_idx in np.arange(num_clusters):
    # finding the euclidean disntance between given centroid and all inputs
    centroid = KMC_Model.cluster_centers_[centroid_idx]
    centroid_matrix = np.matmul(np.ones(X_nparr_normalize.shape[0]).reshape(1,␣
 ↪-1).T, centroid.reshape(1, -1))
    dist.append(np.sum(np.power(centroid_matrix - X_nparr_normalize, 2),␣
 ↪axis=1))

dist = np.array(dist)
```

The group sizes below.

[77]:
```
# The closest centroid for each sample
closest_centroid = np.argmin(dist, axis=0)
```

[78]:
```
[np.count_nonzero(closest_centroid == i) for i in np.arange(num_clusters)]
```

[78]:
```
[3123, 421, 4886, 512, 41]
```

Each centroid has decent amount of data, which is enough to run neural models.

5.2 NN

Due to the randomization nature of the K-Means Clustering Algorithm, the grouping varies every time we run the model. So, we will use a random search for each group's best hyperparameters, including the perception number in the first, second, and third hidden layers.

5.2.1 Model and Training

[79]:
```
def nn_KMC_type(h0, h1, h2):
    model = Sequential()
```

```
    model.add(Dense(h0, input_dim=X_nparr_normalize.shape[1],␣
↪kernel_initializer='normal', activation='relu'))
    model.add(Dense(h1, kernel_initializer='normal', activation='relu'))
    model.add(Dense(h2, kernel_initializer='normal', activation='relu'))
    model.add(Dense(1, kernel_initializer='normal'))
    model.compile(loss='mean_squared_error', optimizer='adam')
    return model
```

[80]:
```
nn_models = []
for group_idx in np.arange(num_clusters):
    print("GROUP {0}".format(group_idx))

    NN_Model = KerasRegressor(build_fn=nn_KMC_type)

    rnd_search_cv = RandomizedSearchCV(
        estimator=NN_Model,
        param_distributions={
            "h0": np.arange(10, 30).tolist(),
            "h1": np.arange(5, 15).tolist(),
            "h2": np.arange(1, 10).tolist()
        },
        n_iter=30,
        cv=3,
        verbose=0,
        n_jobs=-1
    ).fit(
        X_nparr_normalize[closest_centroid == group_idx],
        y_nparr[closest_centroid == group_idx],
        verbose=0,
        epochs=500
    )

    display(rnd_search_cv.best_params_)

    nn_models.append(rnd_search_cv.best_estimator_.model)
```

```
GROUP 0
```

```
2022-12-04 20:43:38.389206: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-12-04 20:43:38.389504: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
```

```
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-12-04 20:43:38.397887: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-12-04 20:43:38.399074: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  SSE4.1 SSE4.2 AVX AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-12-04 20:43:38.520133: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2022-12-04 20:43:38.521351: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency:
2199995000 Hz
2022-12-04 20:43:38.523179: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2022-12-04 20:43:38.526034: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency:
2199995000 Hz
2022-12-04 20:43:38.527564: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2022-12-04 20:43:38.529157: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency:
2199995000 Hz
2022-12-04 20:43:38.540764: I
tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:116] None of the MLIR
optimization passes are enabled (registered 2)
2022-12-04 20:43:38.541498: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:112] CPU Frequency:
2199995000 Hz

33/33 [==============================] - 0s 3ms/step - loss: 1.7641
33/33 [==============================] - 0s 1ms/step - loss: 1.8622
33/33 [==============================] - 0s 2ms/step - loss: 1.8144
33/33 [==============================] - 0s 2ms/step - loss: 1.6966
33/33 [==============================] - 0s 2ms/step - loss: 5.1146
33/33 [==============================] - 0s 3ms/step - loss: 1.7278
33/33 [==============================] - 0s 3ms/step - loss: 1.8950
33/33 [==============================] - 0s 1ms/step - loss: 1.7474
33/33 [==============================] - 0s 3ms/step - loss: 5.0729
```

```
33/33 [==============================] - 0s 1ms/step - loss: 2.0963
33/33 [==============================] - 0s 3ms/step - loss: 5.1151
33/33 [==============================] - 0s 1ms/step - loss: 1.6675
33/33 [==============================] - 0s 3ms/step - loss: 1.9042
33/33 [==============================] - 0s 3ms/step - loss: 1.8155
33/33 [==============================] - 0s 1ms/step - loss: 1.5884
33/33 [==============================] - 0s 3ms/step - loss: 1.7511
33/33 [==============================] - 0s 3ms/step - loss: 1.9399
33/33 [==============================] - 0s 2ms/step - loss: 1.9291
33/33 [==============================] - 0s 1ms/step - loss: 1.7543
33/33 [==============================] - 0s 2ms/step - loss: 1.7805
33/33 [==============================] - 0s 2ms/step - loss: 1.9885
33/33 [==============================] - 0s 2ms/step - loss: 1.9387
33/33 [==============================] - 0s 933us/step - loss: 5.1169
33/33 [==============================] - 0s 1ms/step - loss: 1.7724
33/33 [==============================] - 0s 2ms/step - loss: 1.9146
33/33 [==============================] - 0s 2ms/step - loss: 1.8246
33/33 [==============================] - 0s 3ms/step - loss: 1.6198
33/33 [==============================] - 0s 978us/step - loss: 1.8500
33/33 [==============================] - 0s 3ms/step - loss: 1.8110
33/33 [==============================] - 0s 3ms/step - loss: 1.7755
33/33 [==============================] - 0s 910us/step - loss: 1.8858
33/33 [==============================] - 0s 1ms/step - loss: 1.6789
33/33 [==============================] - 0s 3ms/step - loss: 1.6465
33/33 [==============================] - 0s 2ms/step - loss: 1.7842
33/33 [==============================] - 0s 2ms/step - loss: 1.7244
33/33 [==============================] - 1s 12ms/step - loss: 1.7615
33/33 [==============================] - 0s 1ms/step - loss: 5.0729
33/33 [==============================] - 0s 1ms/step - loss: 1.8160
33/33 [==============================] - 0s 3ms/step - loss: 1.6979
33/33 [==============================] - 0s 2ms/step - loss: 1.8377
33/33 [==============================] - 0s 919us/step - loss: 1.8411
33/33 [==============================] - 0s 2ms/step - loss: 1.5768
33/33 [==============================] - 0s 3ms/step - loss: 1.7837
33/33 [==============================] - 0s 939us/step - loss: 5.1169
33/33 [==============================] - 0s 1ms/step - loss: 1.7115
33/33 [==============================] - 0s 1ms/step - loss: 1.8927
33/33 [==============================] - 0s 1ms/step - loss: 1.7970
33/33 [==============================] - 0s 3ms/step - loss: 1.6530
33/33 [==============================] - 0s 935us/step - loss: 5.1157
33/33 [==============================] - 0s 2ms/step - loss: 1.6754
33/33 [==============================] - 0s 3ms/step - loss: 1.6148
33/33 [==============================] - 0s 1ms/step - loss: 1.7458
33/33 [==============================] - 0s 3ms/step - loss: 1.8107
33/33 [==============================] - 0s 853us/step - loss: 1.4969
33/33 [==============================] - 0s 3ms/step - loss: 1.9148
33/33 [==============================] - 0s 1ms/step - loss: 5.1181
33/33 [==============================] - 0s 2ms/step - loss: 1.9158
```

```
33/33 [==============================] - 0s 2ms/step - loss: 1.6082
33/33 [==============================] - 0s 1ms/step - loss: 1.8009
33/33 [==============================] - 0s 2ms/step - loss: 1.5996
33/33 [==============================] - 0s 3ms/step - loss: 1.7652
33/33 [==============================] - 0s 1ms/step - loss: 5.1146
33/33 [==============================] - 0s 3ms/step - loss: 1.7866
33/33 [==============================] - 0s 3ms/step - loss: 1.7518
33/33 [==============================] - 0s 2ms/step - loss: 1.8383
33/33 [==============================] - 0s 3ms/step - loss: 1.8347
33/33 [==============================] - 0s 2ms/step - loss: 1.8887
33/33 [==============================] - 0s 2ms/step - loss: 1.7385
33/33 [==============================] - 0s 2ms/step - loss: 1.6741
33/33 [==============================] - 0s 2ms/step - loss: 1.7926
33/33 [==============================] - 0s 1ms/step - loss: 2.1815
33/33 [==============================] - 0s 2ms/step - loss: 1.7536
33/33 [==============================] - 0s 3ms/step - loss: 1.8707
33/33 [==============================] - 0s 3ms/step - loss: 1.8613
33/33 [==============================] - 0s 2ms/step - loss: 1.7113
33/33 [==============================] - 0s 1ms/step - loss: 1.7785
33/33 [==============================] - 0s 2ms/step - loss: 1.7958
33/33 [==============================] - 0s 980us/step - loss: 1.6123
33/33 [==============================] - 0s 3ms/step - loss: 1.7662
33/33 [==============================] - 0s 1ms/step - loss: 1.7766
33/33 [==============================] - 0s 1ms/step - loss: 5.0728
33/33 [==============================] - 0s 2ms/step - loss: 1.8544
33/33 [==============================] - 0s 3ms/step - loss: 1.7913
33/33 [==============================] - 0s 2ms/step - loss: 1.7862
33/33 [==============================] - 0s 2ms/step - loss: 1.8920
33/33 [==============================] - 0s 1ms/step - loss: 1.7696
33/33 [==============================] - 0s 940us/step - loss: 1.6637
33/33 [==============================] - 0s 2ms/step - loss: 5.0728
33/33 [==============================] - 0s 909us/step - loss: 5.1148
33/33 [==============================] - 0s 871us/step - loss: 1.6525

{'h2': 6, 'h1': 9, 'h0': 25}

GROUP 1
5/5 [==============================] - 0s 1ms/step - loss: 8.6708
5/5 [==============================] - 0s 1ms/step - loss: 2.9907
5/5 [==============================] - 0s 1ms/step - loss: 3.8670
5/5 [==============================] - 0s 1ms/step - loss: 4.0800
5/5 [==============================] - 0s 1ms/step - loss: 4.0428
5/5 [==============================] - 0s 896us/step - loss: 4.0604
5/5 [==============================] - 0s 929us/step - loss: 3.1181
5/5 [==============================] - 0s 908us/step - loss: 4.0499
5/5 [==============================] - 0s 1ms/step - loss: 4.6320
5/5 [==============================] - 0s 1ms/step - loss: 9.8223
5/5 [==============================] - 0s 877us/step - loss: 8.6705
5/5 [==============================] - 0s 1ms/step - loss: 3.5987
```

```
5/5 [==============================] - 0s 997us/step - loss: 3.7635
5/5 [==============================] - 0s 992us/step - loss: 9.4814
5/5 [==============================] - 0s 1ms/step - loss: 3.6518
5/5 [==============================] - 0s 1ms/step - loss: 3.2387
5/5 [==============================] - 0s 964us/step - loss: 4.1640
5/5 [==============================] - 0s 953us/step - loss: 3.5463
5/5 [==============================] - 0s 1ms/step - loss: 3.6917
5/5 [==============================] - 0s 1ms/step - loss: 3.0628
5/5 [==============================] - 0s 976us/step - loss: 3.7878
5/5 [==============================] - 0s 955us/step - loss: 2.7098
5/5 [==============================] - 0s 972us/step - loss: 3.2022
5/5 [==============================] - 0s 1ms/step - loss: 4.4828
5/5 [==============================] - 0s 951us/step - loss: 3.9285
5/5 [==============================] - 0s 1ms/step - loss: 3.4480
5/5 [==============================] - 0s 1ms/step - loss: 3.4824
5/5 [==============================] - 0s 917us/step - loss: 3.6409
5/5 [==============================] - 0s 948us/step - loss: 8.6701
5/5 [==============================] - 0s 973us/step - loss: 4.4499
5/5 [==============================] - 0s 1ms/step - loss: 3.2146
5/5 [==============================] - 0s 1ms/step - loss: 4.2604
5/5 [==============================] - 0s 846us/step - loss: 3.3987
5/5 [==============================] - 0s 2ms/step - loss: 3.8195
5/5 [==============================] - 0s 2ms/step - loss: 3.5942
5/5 [==============================] - 0s 2ms/step - loss: 3.7387
5/5 [==============================] - 0s 10ms/step - loss: 4.2753
5/5 [==============================] - 0s 1ms/step - loss: 3.5191
5/5 [==============================] - 0s 7ms/step - loss: 2.7316
5/5 [==============================] - 0s 1ms/step - loss: 9.4829
5/5 [==============================] - 0s 2ms/step - loss: 3.7033
5/5 [==============================] - 0s 2ms/step - loss: 3.8702
5/5 [==============================] - 0s 1ms/step - loss: 4.3383
5/5 [==============================] - 0s 1ms/step - loss: 2.8958
5/5 [==============================] - 0s 1ms/step - loss: 3.5918
5/5 [==============================] - 0s 2ms/step - loss: 3.7256
5/5 [==============================] - 0s 1ms/step - loss: 3.6541
5/5 [==============================] - 0s 2ms/step - loss: 2.9778
5/5 [==============================] - 0s 2ms/step - loss: 3.6798
5/5 [==============================] - 0s 1ms/step - loss: 4.5220
5/5 [==============================] - 0s 1ms/step - loss: 9.4823
5/5 [==============================] - 0s 2ms/step - loss: 3.3212
5/5 [==============================] - 0s 2ms/step - loss: 4.1390
5/5 [==============================] - 0s 3ms/step - loss: 9.8191
5/5 [==============================] - 0s 1ms/step - loss: 3.7191
5/5 [==============================] - 0s 8ms/step - loss: 3.7775
5/5 [==============================] - 0s 1ms/step - loss: 3.3780
5/5 [==============================] - 0s 2ms/step - loss: 4.3352
5/5 [==============================] - 0s 1ms/step - loss: 3.1838
5/5 [==============================] - 0s 1ms/step - loss: 4.2790
```

```
5/5 [==============================] - 0s 2ms/step - loss: 4.2646
5/5 [==============================] - 0s 2ms/step - loss: 3.8392
5/5 [==============================] - 0s 1ms/step - loss: 9.4821
5/5 [==============================] - 0s 1ms/step - loss: 3.3117
5/5 [==============================] - 0s 1ms/step - loss: 3.8455
5/5 [==============================] - 0s 1ms/step - loss: 3.0900
5/5 [==============================] - 0s 1ms/step - loss: 8.6731
5/5 [==============================] - 0s 1ms/step - loss: 9.4841
5/5 [==============================] - 0s 1ms/step - loss: 3.0022
5/5 [==============================] - 0s 994us/step - loss: 3.6786
5/5 [==============================] - 0s 11ms/step - loss: 3.1232
5/5 [==============================] - 0s 1ms/step - loss: 4.3081
5/5 [==============================] - 0s 1ms/step - loss: 3.8833
5/5 [==============================] - 0s 12ms/step - loss: 4.0882
5/5 [==============================] - 0s 1ms/step - loss: 2.9069
5/5 [==============================] - 0s 1ms/step - loss: 3.7796
5/5 [==============================] - 0s 1ms/step - loss: 4.1886
5/5 [==============================] - 0s 1ms/step - loss: 3.5878
5/5 [==============================] - 0s 1ms/step - loss: 2.9066
5/5 [==============================] - 0s 1ms/step - loss: 3.5697
5/5 [==============================] - 0s 1ms/step - loss: 9.8220
5/5 [==============================] - 0s 1ms/step - loss: 3.7342
5/5 [==============================] - 0s 1ms/step - loss: 4.3735
5/5 [==============================] - 0s 1ms/step - loss: 3.0340
5/5 [==============================] - 0s 968us/step - loss: 4.5682
5/5 [==============================] - 0s 949us/step - loss: 4.1412
5/5 [==============================] - 0s 1ms/step - loss: 3.3874
5/5 [==============================] - 0s 1ms/step - loss: 3.6394
5/5 [==============================] - 0s 954us/step - loss: 8.6701
5/5 [==============================] - 0s 902us/step - loss: 3.4096

{'h2': 8, 'h1': 6, 'h0': 19}

GROUP 2
51/51 [==============================] - 0s 2ms/step - loss: 1.1780
51/51 [==============================] - 0s 2ms/step - loss: 1.1854
51/51 [==============================] - 0s 1ms/step - loss: 1.1507
51/51 [==============================] - 0s 2ms/step - loss: 1.1092
51/51 [==============================] - 0s 2ms/step - loss: 1.2168
51/51 [==============================] - 0s 2ms/step - loss: 1.1257
51/51 [==============================] - 0s 2ms/step - loss: 1.1351
51/51 [==============================] - 0s 2ms/step - loss: 1.1615
51/51 [==============================] - 0s 2ms/step - loss: 1.1197
51/51 [==============================] - 0s 2ms/step - loss: 1.2136
51/51 [==============================] - 0s 2ms/step - loss: 1.1263
51/51 [==============================] - 0s 907us/step - loss: 1.1777
51/51 [==============================] - 0s 2ms/step - loss: 1.1710
51/51 [==============================] - 0s 2ms/step - loss: 1.2012
51/51 [==============================] - 0s 2ms/step - loss: 1.1573
```

```
51/51 [==============================] - 0s 2ms/step - loss: 3.3545
51/51 [==============================] - 0s 2ms/step - loss: 1.1878
51/51 [==============================] - 0s 2ms/step - loss: 1.2264
51/51 [==============================] - 0s 2ms/step - loss: 1.1829
51/51 [==============================] - 0s 2ms/step - loss: 1.1502
51/51 [==============================] - 0s 3ms/step - loss: 1.1775
51/51 [==============================] - 0s 2ms/step - loss: 1.2125
51/51 [==============================] - 0s 2ms/step - loss: 1.1042
51/51 [==============================] - 1s 5ms/step - loss: 3.2635
51/51 [==============================] - 0s 2ms/step - loss: 1.2582
51/51 [==============================] - 0s 2ms/step - loss: 1.1720
51/51 [==============================] - 0s 2ms/step - loss: 1.1921
51/51 [==============================] - 0s 1ms/step - loss: 1.1950
51/51 [==============================] - 0s 2ms/step - loss: 1.1483
51/51 [==============================] - 0s 2ms/step - loss: 1.1614
51/51 [==============================] - 0s 2ms/step - loss: 1.2828
51/51 [==============================] - 0s 2ms/step - loss: 1.2049
51/51 [==============================] - 0s 2ms/step - loss: 1.1817
51/51 [==============================] - 0s 2ms/step - loss: 1.1972
51/51 [==============================] - 0s 2ms/step - loss: 1.1340
51/51 [==============================] - 0s 1ms/step - loss: 1.2913
51/51 [==============================] - 0s 2ms/step - loss: 3.3540
51/51 [==============================] - 0s 1ms/step - loss: 1.2416
51/51 [==============================] - 0s 2ms/step - loss: 1.1893
51/51 [==============================] - 0s 2ms/step - loss: 1.1878
51/51 [==============================] - 0s 3ms/step - loss: 1.2352
51/51 [==============================] - 0s 3ms/step - loss: 1.2204
51/51 [==============================] - 0s 1ms/step - loss: 1.1404
51/51 [==============================] - 0s 3ms/step - loss: 1.1311
51/51 [==============================] - 0s 2ms/step - loss: 1.2089
51/51 [==============================] - 0s 2ms/step - loss: 1.1513
51/51 [==============================] - 0s 2ms/step - loss: 1.2343
51/51 [==============================] - 0s 2ms/step - loss: 1.1898
51/51 [==============================] - 0s 2ms/step - loss: 1.2146
51/51 [==============================] - 0s 1ms/step - loss: 1.2140
51/51 [==============================] - 0s 2ms/step - loss: 1.1838
51/51 [==============================] - 0s 2ms/step - loss: 1.1493
51/51 [==============================] - 0s 2ms/step - loss: 1.3063
51/51 [==============================] - 0s 2ms/step - loss: 1.1868
51/51 [==============================] - 0s 2ms/step - loss: 1.1272
51/51 [==============================] - 0s 1ms/step - loss: 1.2154
51/51 [==============================] - 0s 1ms/step - loss: 1.1242
51/51 [==============================] - 0s 2ms/step - loss: 1.2079
51/51 [==============================] - 0s 2ms/step - loss: 1.2887
51/51 [==============================] - 0s 2ms/step - loss: 1.1048
51/51 [==============================] - 0s 2ms/step - loss: 1.1896
51/51 [==============================] - 0s 2ms/step - loss: 1.1524
51/51 [==============================] - 0s 2ms/step - loss: 1.1174
```

```
51/51 [==============================] - 0s 2ms/step - loss: 3.3543
51/51 [==============================] - 0s 2ms/step - loss: 1.1639
51/51 [==============================] - 0s 2ms/step - loss: 1.1899
51/51 [==============================] - 0s 2ms/step - loss: 1.1253
51/51 [==============================] - 1s 4ms/step - loss: 1.2192
51/51 [==============================] - 0s 2ms/step - loss: 1.2619
51/51 [==============================] - 0s 2ms/step - loss: 1.1402
51/51 [==============================] - 0s 2ms/step - loss: 1.1954
51/51 [==============================] - 0s 2ms/step - loss: 1.1316
51/51 [==============================] - 0s 2ms/step - loss: 1.1295
51/51 [==============================] - 0s 2ms/step - loss: 1.2016
51/51 [==============================] - 0s 2ms/step - loss: 1.2745
51/51 [==============================] - 0s 903us/step - loss: 1.1331
51/51 [==============================] - 0s 907us/step - loss: 1.1653
51/51 [==============================] - 0s 2ms/step - loss: 1.1468
51/51 [==============================] - 0s 2ms/step - loss: 1.1793
51/51 [==============================] - 0s 2ms/step - loss: 1.1411
51/51 [==============================] - 0s 2ms/step - loss: 1.1581
51/51 [==============================] - 0s 2ms/step - loss: 3.3546
51/51 [==============================] - 0s 2ms/step - loss: 1.1992
51/51 [==============================] - 0s 2ms/step - loss: 1.2422
51/51 [==============================] - 0s 3ms/step - loss: 1.1680
51/51 [==============================] - 0s 2ms/step - loss: 1.1727
51/51 [==============================] - 0s 2ms/step - loss: 1.1560
51/51 [==============================] - 0s 877us/step - loss: 1.0571
51/51 [==============================] - 0s 1ms/step - loss: 1.1672
51/51 [==============================] - 0s 932us/step - loss: 3.2633

{'h2': 5, 'h1': 12, 'h0': 15}

GROUP 3
6/6 [==============================] - 0s 2ms/step - loss: 1.2174
6/6 [==============================] - 0s 1ms/step - loss: 2.4439
6/6 [==============================] - 0s 1ms/step - loss: 1.2903
6/6 [==============================] - 0s 1ms/step - loss: 2.3560
6/6 [==============================] - 0s 1ms/step - loss: 1.3648
6/6 [==============================] - 0s 1ms/step - loss: 2.3649
6/6 [==============================] - 0s 1ms/step - loss: 3.1471
6/6 [==============================] - 0s 2ms/step - loss: 1.2872
6/6 [==============================] - 0s 1ms/step - loss: 3.6379
6/6 [==============================] - 0s 1ms/step - loss: 2.4178
6/6 [==============================] - 0s 1ms/step - loss: 1.2960
6/6 [==============================] - 0s 1ms/step - loss: 1.5226
6/6 [==============================] - 0s 1ms/step - loss: 2.4351
6/6 [==============================] - 0s 1ms/step - loss: 1.3133
6/6 [==============================] - 0s 1ms/step - loss: 1.3182
6/6 [==============================] - 0s 2ms/step - loss: 2.3329
6/6 [==============================] - 0s 9ms/step - loss: 2.3944
6/6 [==============================] - 0s 1ms/step - loss: 1.3637
```

```
6/6 [==============================] - 0s 1ms/step - loss: 1.2648
6/6 [==============================] - 0s 1ms/step - loss: 1.3199
6/6 [==============================] - 0s 10ms/step - loss: 3.6391
6/6 [==============================] - 0s 1ms/step - loss: 2.4565
6/6 [==============================] - 0s 1ms/step - loss: 1.3165
6/6 [==============================] - 0s 1ms/step - loss: 1.2375
6/6 [==============================] - 0s 1ms/step - loss: 2.5233
6/6 [==============================] - 0s 1ms/step - loss: 1.2353
6/6 [==============================] - 0s 1ms/step - loss: 1.2633
6/6 [==============================] - 0s 1ms/step - loss: 2.5155
6/6 [==============================] - 0s 1ms/step - loss: 2.4766
6/6 [==============================] - 0s 1ms/step - loss: 3.1471
6/6 [==============================] - 0s 2ms/step - loss: 1.1771
6/6 [==============================] - 0s 1ms/step - loss: 1.2584
6/6 [==============================] - 0s 1ms/step - loss: 3.6394
6/6 [==============================] - 0s 1ms/step - loss: 2.5606
6/6 [==============================] - 0s 981us/step - loss: 3.6394
6/6 [==============================] - 0s 904us/step - loss: 3.1470
6/6 [==============================] - 0s 1ms/step - loss: 2.4804
6/6 [==============================] - 0s 1ms/step - loss: 1.3145
6/6 [==============================] - 0s 1ms/step - loss: 2.4636
6/6 [==============================] - 0s 1ms/step - loss: 1.2912
6/6 [==============================] - 0s 980us/step - loss: 1.3469
6/6 [==============================] - 0s 1ms/step - loss: 1.9670
6/6 [==============================] - 0s 1ms/step - loss: 1.3128
6/6 [==============================] - 0s 1ms/step - loss: 1.2899
6/6 [==============================] - 0s 1ms/step - loss: 1.1716
6/6 [==============================] - 0s 1ms/step - loss: 1.4157
6/6 [==============================] - 0s 1ms/step - loss: 2.3731
6/6 [==============================] - 0s 1ms/step - loss: 1.3180
6/6 [==============================] - 0s 1ms/step - loss: 2.2358
6/6 [==============================] - 0s 1ms/step - loss: 1.2203
6/6 [==============================] - 0s 943us/step - loss: 1.2299
6/6 [==============================] - 0s 1ms/step - loss: 3.7663
6/6 [==============================] - 0s 1ms/step - loss: 3.1471
6/6 [==============================] - 0s 1ms/step - loss: 2.5976
6/6 [==============================] - 0s 1ms/step - loss: 3.6395
6/6 [==============================] - 0s 1ms/step - loss: 1.2912
6/6 [==============================] - 0s 912us/step - loss: 1.2179
6/6 [==============================] - 0s 1ms/step - loss: 2.5921
6/6 [==============================] - 0s 1ms/step - loss: 1.2018
6/6 [==============================] - 0s 1ms/step - loss: 1.3245
6/6 [==============================] - 0s 1ms/step - loss: 3.7662
6/6 [==============================] - 0s 930us/step - loss: 3.1471
6/6 [==============================] - 0s 877us/step - loss: 1.1546
6/6 [==============================] - 0s 1ms/step - loss: 2.5356
6/6 [==============================] - 0s 1ms/step - loss: 3.1471
6/6 [==============================] - 0s 1ms/step - loss: 1.3358
```

```
6/6 [==============================] - 0s 1ms/step - loss: 2.5402
6/6 [==============================] - 0s 1ms/step - loss: 1.2757
6/6 [==============================] - 0s 1ms/step - loss: 1.2929
6/6 [==============================] - 0s 1ms/step - loss: 1.2920
6/6 [==============================] - 0s 987us/step - loss: 2.3072
6/6 [==============================] - 0s 1ms/step - loss: 1.2398
6/6 [==============================] - 0s 1ms/step - loss: 2.2963
6/6 [==============================] - 0s 9ms/step - loss: 1.2964
6/6 [==============================] - 0s 1ms/step - loss: 1.1915
6/6 [==============================] - 0s 1ms/step - loss: 2.5121
6/6 [==============================] - 0s 1ms/step - loss: 1.2626
6/6 [==============================] - 0s 955us/step - loss: 2.4789
6/6 [==============================] - 0s 1ms/step - loss: 1.3671
6/6 [==============================] - 0s 1ms/step - loss: 1.3112
6/6 [==============================] - 0s 1ms/step - loss: 1.2077
6/6 [==============================] - 0s 1ms/step - loss: 3.7660
6/6 [==============================] - 0s 994us/step - loss: 1.2796
6/6 [==============================] - 0s 929us/step - loss: 1.1811
6/6 [==============================] - 0s 2ms/step - loss: 2.2233
6/6 [==============================] - 0s 2ms/step - loss: 1.2545
6/6 [==============================] - 0s 1ms/step - loss: 1.1411
6/6 [==============================] - 0s 1ms/step - loss: 2.0536
6/6 [==============================] - 0s 2ms/step - loss: 1.2833
6/6 [==============================] - 0s 2ms/step - loss: 1.2965

{'h2': 8, 'h1': 7, 'h0': 26}

GROUP 4
1/1 [==============================] - 0s 145ms/step - loss: 5.1521
1/1 [==============================] - 0s 74ms/step - loss: 12.3321
1/1 [==============================] - 0s 72ms/step - loss: 1.6010
1/1 [==============================] - 0s 79ms/step - loss: 4.9502
1/1 [==============================] - 0s 70ms/step - loss: 2.2718
1/1 [==============================] - 0s 72ms/step - loss: 10.3668
1/1 [==============================] - 0s 141ms/step - loss: 9.5877
1/1 [==============================] - 0s 94ms/step - loss: 5.1004

WARNING:tensorflow:5 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c40609d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5dd008c9d0> triggered
```

```
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08d609da60> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:5 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f10ec6a71f0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 114ms/step - loss: 2.1882
1/1 [==============================] - 0s 72ms/step - loss: 8.4514
1/1 [==============================] - 0s 76ms/step - loss: 2.2454
1/1 [==============================] - 0s 87ms/step - loss: 4.9467

WARNING:tensorflow:6 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f1104645d30> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa684725940> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
```

```
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b0509160> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da445e9d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 102ms/step - loss: 4.9938
1/1 [==============================] - 0s 99ms/step - loss: 5.0458
1/1 [==============================] - 0s 96ms/step - loss: 2.4155
1/1 [==============================] - 0s 94ms/step - loss: 9.2981

WARNING:tensorflow:6 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c40601f0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08887d0310> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
```

```
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:6 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da41d5f70> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 166ms/step - loss: 1.6571
1/1 [==============================] - 0s 98ms/step - loss: 5.8259
1/1 [==============================] - 0s 100ms/step - loss: 9.5877

WARNING:tensorflow:6 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f111c042ee0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 91ms/step - loss: 12.8430

WARNING:tensorflow:7 out of the last 12 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac344d30> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:7 out of the last 12 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b0466040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
```

```
WARNING:tensorflow:7 out of the last 12 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da461c790> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 131ms/step - loss: 8.8136
1/1 [==============================] - 0s 108ms/step - loss: 4.8789
1/1 [==============================] - 0s 104ms/step - loss: 14.6238

WARNING:tensorflow:7 out of the last 12 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11146e8040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 79ms/step - loss: 3.2795

WARNING:tensorflow:8 out of the last 13 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b05d0dc0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:8 out of the last 13 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5d887edd30> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 167ms/step - loss: 10.0052
```

```
1/1 [==============================] - 0s 165ms/step - loss: 2.2548

WARNING:tensorflow:8 out of the last 13 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c4747ee0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:8 out of the last 13 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11147e9af0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 91ms/step - loss: 3.9227
1/1 [==============================] - 0s 90ms/step - loss: 4.8037

WARNING:tensorflow:9 out of the last 14 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c4628040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:9 out of the last 14 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11140bb0d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:9 out of the last 14 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08d0be4e50> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
```

```
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:9 out of the last 14 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da461c700> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 153ms/step - loss: 5.5286
1/1 [==============================] - 0s 103ms/step - loss: 9.5877
1/1 [==============================] - 0s 112ms/step - loss: 9.1594
1/1 [==============================] - 0s 88ms/step - loss: 2.4097

WARNING:tensorflow:10 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c4707af0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:10 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b0647040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:10 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f1104785ee0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
```

```
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:10 out of the last 15 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5dd3da1940> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 156ms/step - loss: 2.1353
1/1 [==============================] - 0s 111ms/step - loss: 14.1296
1/1 [==============================] - 0s 74ms/step - loss: 4.3615
1/1 [==============================] - 0s 81ms/step - loss: 2.7749

WARNING:tensorflow:11 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa68463c0d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b0042700> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f110450f0d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
```

```
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 16 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5dd008cdc0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 117ms/step - loss: 15.0928
1/1 [==============================] - 0s 112ms/step - loss: 12.5883
1/1 [==============================] - 0s 105ms/step - loss: 2.1815
1/1 [==============================] - 0s 81ms/step - loss: 4.8115

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac72cee0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08d0be4280> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f110450f5e0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da47b3f70> triggered
```

```
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 157ms/step - loss: 14.5880
1/1 [==============================] - 0s 109ms/step - loss: 2.5197
1/1 [==============================] - 0s 74ms/step - loss: 4.8237
1/1 [==============================] - 0s 91ms/step - loss: 14.2719

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6847d30d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08cc050040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11046dd9d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5dbc1971f0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
```

```
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 162ms/step - loss: 2.1837
1/1 [==============================] - 0s 142ms/step - loss: 5.0738
1/1 [==============================] - 0s 96ms/step - loss: 8.5107
1/1 [==============================] - 0s 90ms/step - loss: 2.0461

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6847d35e0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b0733310> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11046dd040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 155ms/step - loss: 4.8598
1/1 [==============================] - 0s 148ms/step - loss: 17.6188
1/1 [==============================] - 0s 104ms/step - loss: 3.1173

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da41b40d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
```

```
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 115ms/step - loss: 4.8439

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac5179d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08cc050a60> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 118ms/step - loss: 8.7304
1/1 [==============================] - 0s 134ms/step - loss: 2.2898

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da41b45e0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 144ms/step - loss: 8.3131

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11041009d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
```

```
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 89ms/step - loss: 4.9703

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b055ea60> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac517040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 125ms/step - loss: 5.0751
1/1 [==============================] - 0s 160ms/step - loss: 2.5849

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da47d19d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 98ms/step - loss: 14.7703

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11041005e0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
```

```
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 100ms/step - loss: 2.6900

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b056bd30> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 146ms/step - loss: 5.4736
1/1 [==============================] - 0s 119ms/step - loss: 2.0328
1/1 [==============================] - 0s 106ms/step - loss: 8.6014

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da47d1040> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac10f9d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f110411b820> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
```

```
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 93ms/step - loss: 5.3763

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08bc6e9ee0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac10f5e0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 106ms/step - loss: 8.4732
1/1 [==============================] - 0s 114ms/step - loss: 4.9292

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f111e4fcca0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5d886bf160> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
```

```
1/1 [==============================] - 0s 88ms/step - loss: 9.5877
1/1 [==============================] - 0s 87ms/step - loss: 2.1943

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac062820> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08bc6e9670> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 184ms/step - loss: 8.8136
1/1 [==============================] - 0s 129ms/step - loss: 17.7167

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11047851f0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5d886bf0d0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 98ms/step - loss: 6.7295
1/1 [==============================] - 0s 76ms/step - loss: 8.8136
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6cec13ca0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b056be50> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 128ms/step - loss: 5.0354
1/1 [==============================] - 0s 156ms/step - loss: 17.7437

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f11147e9b80> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5dd01b6f70> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 113ms/step - loss: 8.8137
1/1 [==============================] - 0s 99ms/step - loss: 5.0478

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac72c1f0> triggered
```

```
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b0375c10> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f111c042dc0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 163ms/step - loss: 12.1228
1/1 [==============================] - 0s 130ms/step - loss: 2.2879
1/1 [==============================] - 0s 113ms/step - loss: 4.9191

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da47d1430> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 82ms/step - loss: 13.7811

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b039eb80> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
```

```
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c4707b80> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f1104034940> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 109ms/step - loss: 4.8900
1/1 [==============================] - 0s 211ms/step - loss: 2.2835
1/1 [==============================] - 0s 105ms/step - loss: 8.7456

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5dd008caf0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 105ms/step - loss: 2.7330

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08b06473a0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
```

```
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f111c042ca0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6c4747dc0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 159ms/step - loss: 5.0485
1/1 [==============================] - 0s 101ms/step - loss: 17.4564
1/1 [==============================] - 0s 102ms/step - loss: 2.2284

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f5da40af5e0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 78ms/step - loss: 5.3934

WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7f08d0be4ca0> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
```

```
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.
WARNING:tensorflow:11 out of the last 11 calls to <function
Model.make_test_function.<locals>.test_function at 0x7fa6ac134940> triggered
tf.function retracing. Tracing is expensive and the excessive number of tracings
could be due to (1) creating @tf.function repeatedly in a loop, (2) passing
tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2),
@tf.function has experimental_relax_shapes=True option that relaxes argument
shapes that can avoid unnecessary retracing. For (3), please refer to
https://www.tensorflow.org/guide/function#controlling_retracing and
https://www.tensorflow.org/api_docs/python/tf/function for  more details.

1/1 [==============================] - 0s 80ms/step - loss: 8.3750
1/1 [==============================] - 0s 78ms/step - loss: 2.1947

{'h2': 7, 'h1': 11, 'h0': 19}
```

5.2.2 Validation

[81]:
```
dist = []
for centroid_idx in np.arange(num_clusters):
    # finding the euclidean disntance between given centroid and all inputs
    centroid = KMC_Model.cluster_centers_[centroid_idx]
    centroid_matrix = np.matmul(np.ones(X_validation_normalized.shape[0]).
 ↪reshape(1, -1).T, centroid.reshape(1, -1))
    dist.append(np.sum(np.power(centroid_matrix - X_validation_normalized, 2),␣
 ↪axis=1))

dist = np.array(dist)
```

[82]:
```
closest_centroid = np.argmin(dist, axis=0)
```

[83]:
```
centroid_sizes = [np.count_nonzero(closest_centroid == i) for i in np.
 ↪arange(num_clusters)]
print("The group sizes are: {0}".format(centroid_sizes))
```

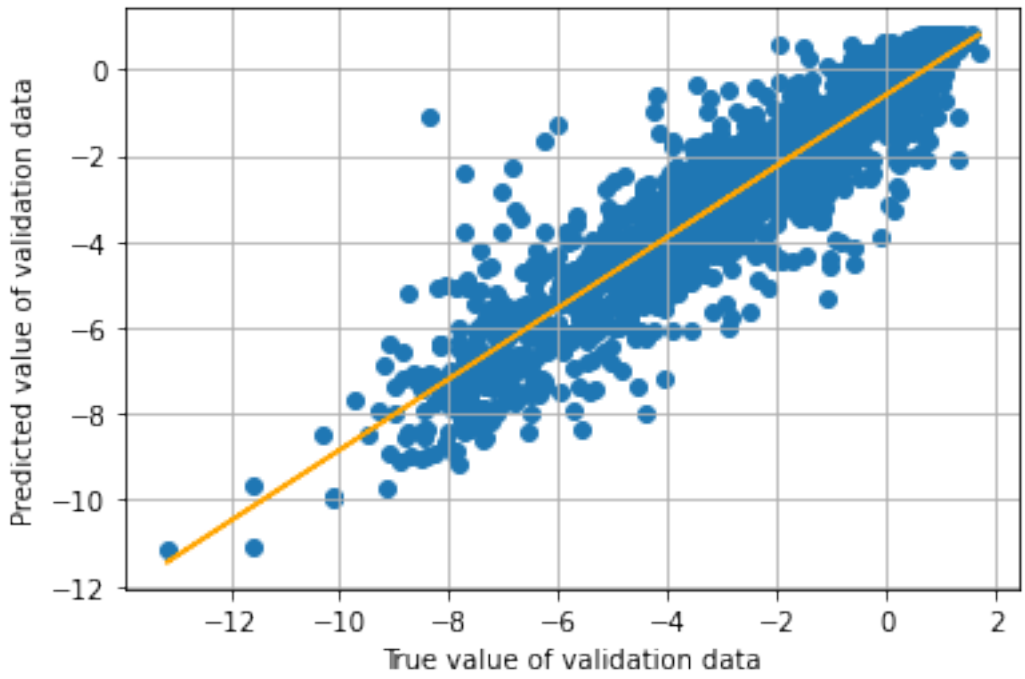
```
The group sizes are: [655, 76, 968, 88, 10]
```

[84]:
```
y_validation_ = np.ones(y_validation.shape)
for group_idx in np.arange(num_clusters):
    y_validation_group = nn_models[group_idx].
 ↪predict(X_validation_normalized[closest_centroid == group_idx])
    y_validation_[closest_centroid == group_idx] = y_validation_group.
 ↪reshape(-1)
```

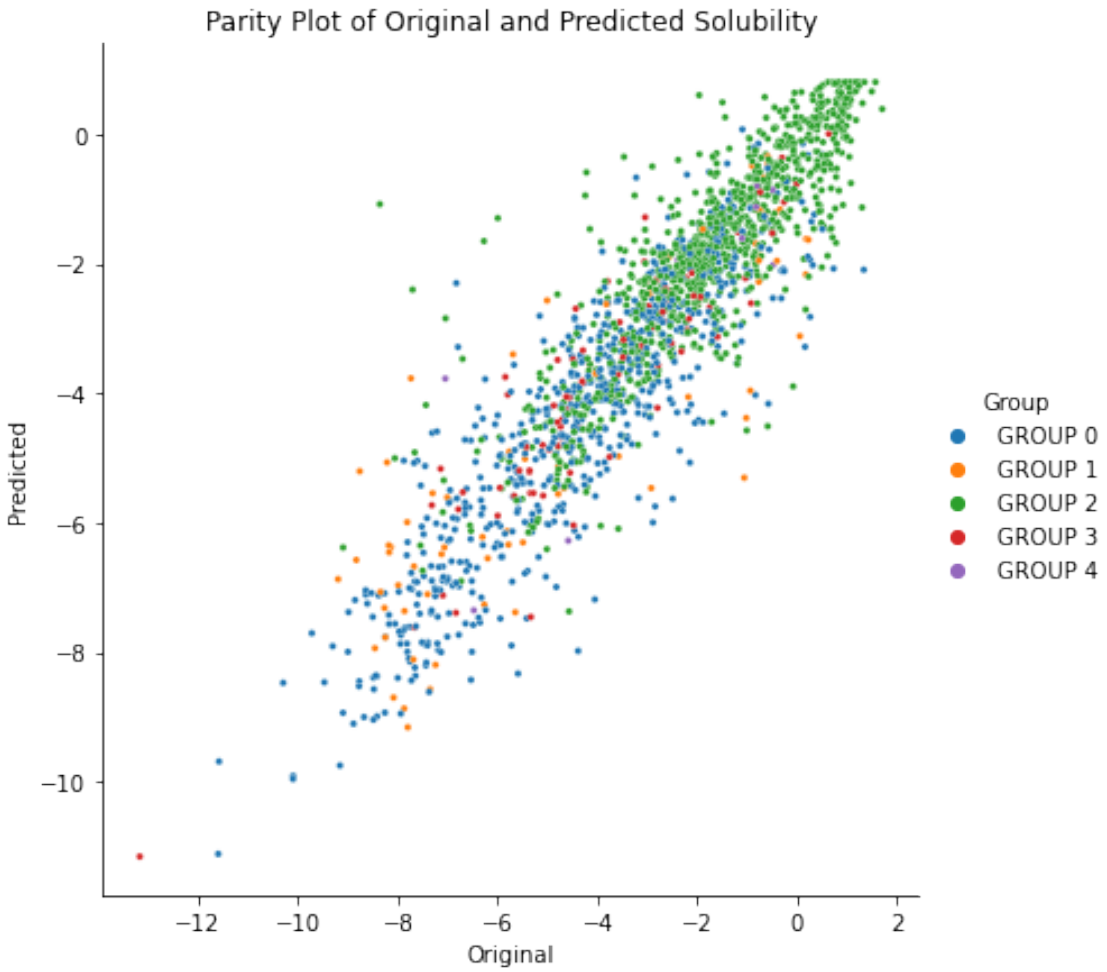
```
    print("GROUP {0}: MSE = {1}".format(group_idx,␣
 ↪mean_squared_error(y_validation[closest_centroid == group_idx],␣
 ↪y_validation_group)))

print("\nOverall: MSE = {0}".format(mean_squared_error(y_validation,␣
 ↪y_validation_)))
```

```
GROUP 0: MSE = 1.1497209739366492
GROUP 1: MSE = 2.417122239192238
GROUP 2: MSE = 0.8943127135070769
GROUP 3: MSE = 0.7364615207843883
GROUP 4: MSE = 1.7443181718756513

Overall: MSE = 1.0488119256147739
```

[85]:
```
y_validation_predict = y_validation_
plt.scatter(y_validation, y_validation_predict)

# line of best fit (linear regression) on the parity plot.
coeff = np.polyfit(y_validation, y_validation_predict, 1)
y_validation_predict_ = np.polyval(coeff, y_validation)
plt.plot(y_validation, y_validation_predict_, c = 'orange')  # parity_plot␣
 ↪(true_data, predicted_data)

plt.xlabel("True value of validation data")
plt.ylabel("Predicted value of validation data")
plt.grid()
plt.show()
```



[86]:
```
df_sns = pd.DataFrame(
    {"Original": y_validation, "Predicted": y_validation_, "Group": ["GROUP␣
 ↪{0}".format(i) for i in closest_centroid]}
)

g = sns.relplot(
    data=df_sns,
    x="Original",
    y="Predicted",
    hue="Group",
    kind="scatter",
    hue_order=["GROUP {0}".format(i) for i in np.arange(num_clusters)],
    s=10,
    aspect=1,
    height=6
)
g.ax.set_title("Parity Plot of Original and Predicted Solubility")
plt.show()
```



In the plot above, we can distinguish each group, which is a good indicator that models aggregate similar samples and neural networks are able to predict their solubility.

[87]:
```
print("Variance of the KMC+NN model is {0}".format(r2_score(y_validation,␣
↪y_validation_)))
```

```
Variance of the KMC+NN model is 0.8252300732023183
```

However, the variance and mean-square-value of the validation data shows that KMC+NN does not provide clear advantage over regular NN model.

[104]:
```
# save model for testing
KMC_NN_model = {}
KMC_NN_model["KMC"] = KMC_Model
KMC_NN_model["nn"] = nn_models

# KMC+NN model predicting precedure
def KMC_NN_predict(KMC_NN_model, X_test):
```

```
    dist = []
    for centroid_idx in np.arange(num_clusters):
        centroid = KMC_NN_model["KMC"].cluster_centers_[centroid_idx]
        centroid_matrix = np.matmul(np.ones(X_test.shape[0]).reshape(1, -1).T,␣
↪centroid.reshape(1, -1))
        dist.append(np.sum(np.power(centroid_matrix - X_test, 2), axis=1))
    closest_centroid = np.argmin(np.array(dist), axis=0)
    y_test_ = np.ones(X_test.shape[0])
    for group_idx in np.arange(num_clusters):
        y_test_group = KMC_NN_model["nn"][group_idx].
↪predict(X_test[closest_centroid == group_idx])
        y_test_[closest_centroid == group_idx] = y_test_group.reshape(-1)
    return y_test_
```

6 Testing and Benchmark

In the previous sections, we have developed the following algorithms to predict the model solubilities. - Linear Regression - Neural Network - Neural Network on Reduced-Dimension Data by Principal Component Analysis (PCA+NN) - Nerual Network on Grouped Data by K-Mean Clusering (KMC+NN)

Now, we are run each model on the testing dataset. The test criteria are following: - Mean Square Error (MSE) - Variance ("r2 score") - Parity Plot

Testing dataset has not expose to any of the model yet, therefore, it serves as an objective benchmark for each model's performace.

[105]:
```
# load check point

# only import testing data
_, _, _, _, X_test, y_test = data_directory.load_data("Solubility Data Splitted.
↪pkl")

# normalize testing data
X_test_normalized = X_scaler.transform(X_test)
```

[106]:
```
def print_testing_metrics(a_true, a_fit):
    print("Variance: {0}".format(r2_score(a_true, a_fit)))
    print("     MSE: {0}".format(mean_squared_error(a_true, a_fit)))
    plt.scatter(a_true, a_fit)

    # line of best fit (linear regression) on the parity plot.
    a_fit_ = np.polyval(np.polyfit(a_true, a_fit, 1), a_true)
    plt.plot(a_true, a_fit_, c = 'orange')

    plt.xlabel("True value of testing data")
    plt.ylabel("Predicted value of testing data")
```

```
    plt.grid()
    plt.show()
```

6.1 Linear Regression

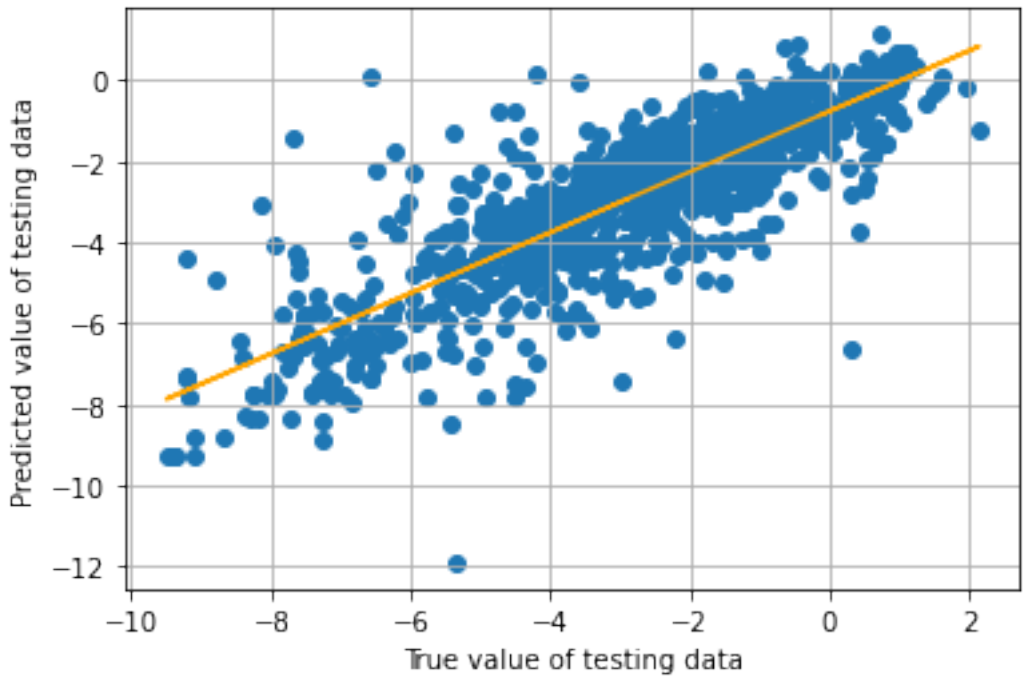
[107]:
```
print_testing_metrics(y_test, linear_regression_model.
↪predict(X_test_normalized))
```

```
Variance: 0.47483327764258265
     MSE: 2.7255157441953415
```

6.2 Nerual Network

[108]:
```
print_testing_metrics(y_test, neural_network_model.predict(X_test_normalized))
```

```
Variance: 0.6934348137447318
     MSE: 1.591015207532998
```



6.3 PCA+NN

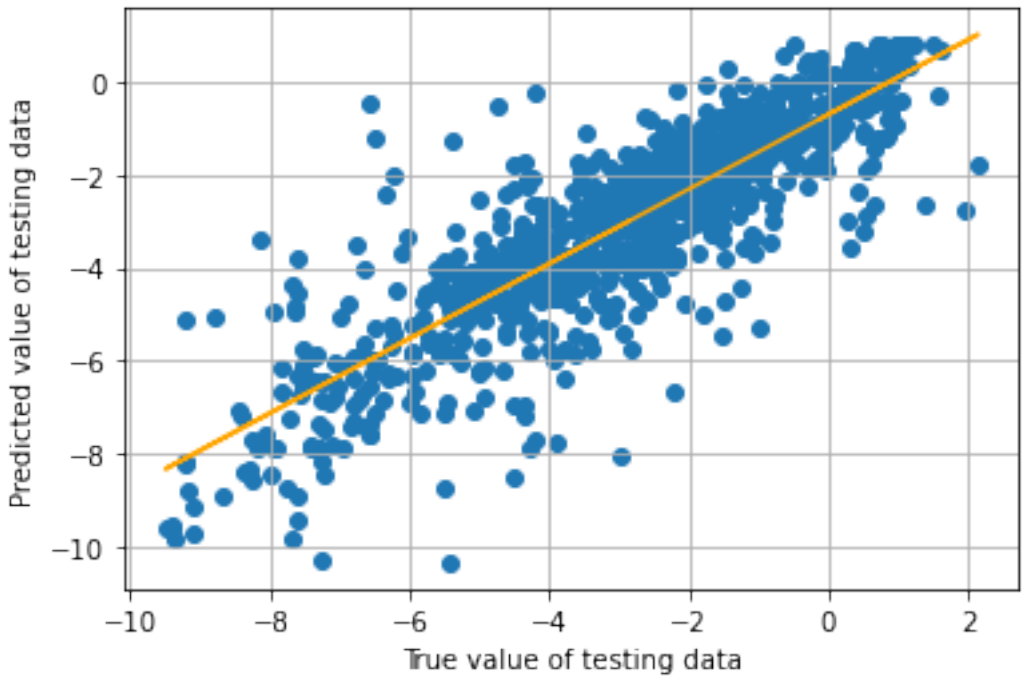
[109]:
```
print_testing_metrics(y_test, PCA_NN_model["nn"].predict(PCA_NN_model["PCA"].
↪transform(X_test_normalized)))
```

```
Variance: 0.6510250618729956
     MSE: 1.8111137810202342
```

6.4 KMC+NN

[110]:
```
print_testing_metrics(y_test, KMC_NN_predict(KMC_NN_model, X_test_normalized))
```

```
Variance: 0.7237957520357781
     MSE: 1.4334476926886888
```



7 Acknowledgement

Thank you to Instructors and TAs for their instruction during the quarter and guide us through the project.

Resource: * Python: Feature/Variable importance after a PCA analysis https://pyquestions.com/feature-variable-importance-after-a-pca-analysis * Keras API reference https://keras.io/api/