kaggle Challenge: House Prices: Advanced Regression Techniques

Abstract:

This report describes an ANN based approach to solve the 'House Prices: Advanced Regression Techniques' challenge in kaggle.This problem is a regression problem where a model has to predict the price of a house based on some given features. Rather than building a model first, here initially the dataset has been preprocessed to create better predictions.Some of them are-

- Target variable skew reduction
- Correlation analysis of features
- Scaling
- One hot encoder
- Missing value handling
- Feature adding
- Attributes skew reduction
  Each processing step has been explained thoroughly in this report with graphs and other visualizing tools. The hyperparameters have been tuned accordingly. Tuning steps and decisions have been discussed thoroughly.

Used libraries:

1. pandas
2. numpy
3. seaborn
4. matplotlib
5. tensorflow
6. sklearn

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn
import tensorflow as tf
```

In [2]:
```
# Reading data from folder named 'data'
train = pd.read_csv("data/train.csv")
test = pd.read_csv("data/test.csv")
```

1. Problem analysis:

From basic analysis, we can give a short description of the problem and dataset:
Problem type: Regression
Training dataset size: 1460
Testing dataset size: 1459

2. Data analysis:

Data analysis starts here:

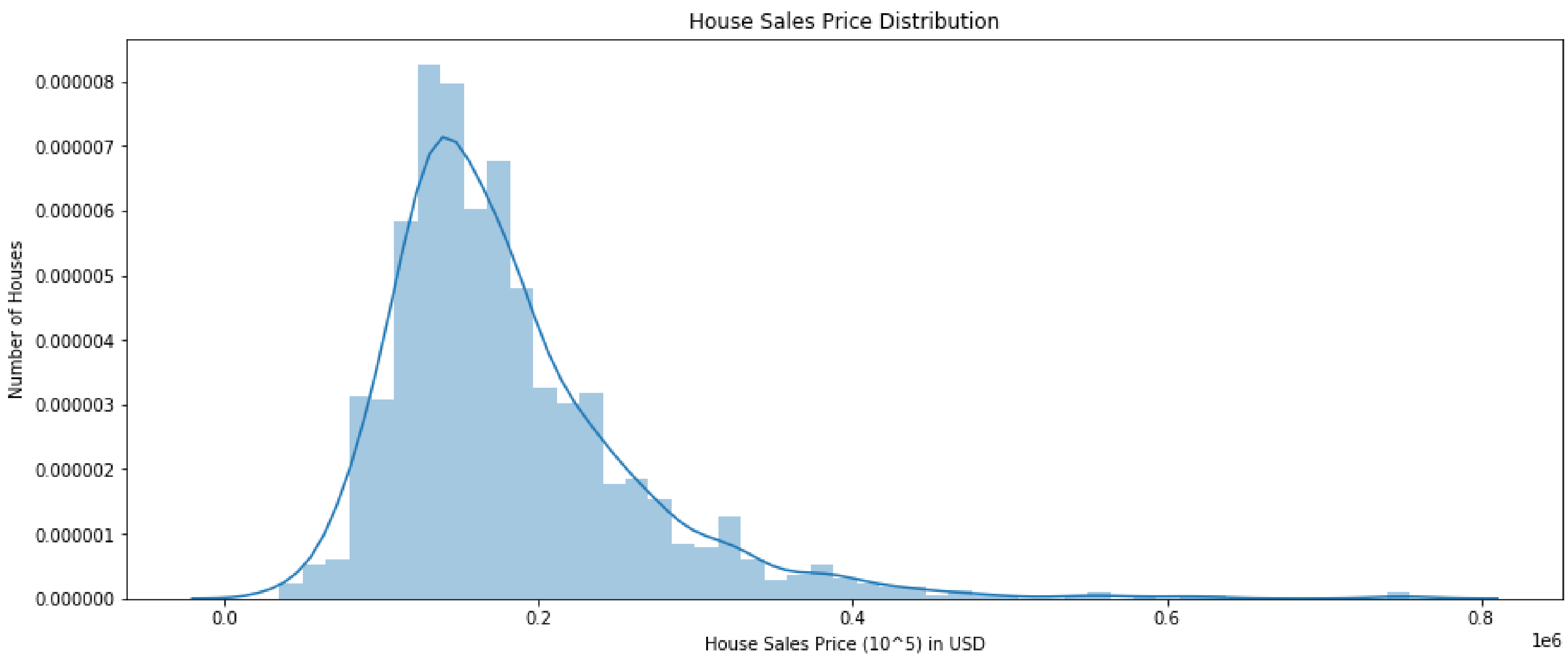
In [3]:
```
#visualizing distribution of target variable

plt.figure(figsize=(15,6))
sns.distplot(train.SalePrice)
plt.ticklabel_format(style='sci', axis='x', scilimits=(0,1))
plt.xlabel("House Sales Price (10^5) in USD")
plt.ylabel("Number of Houses")
plt.title("House Sales Price Distribution")

print("Skew is", train.SalePrice.skew())
```

```
c:\python35\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multid
imensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be in
terpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different resul
t.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Skew is 1.8828757597682129
```



2.1. Target variable analysis:

2.1.1. Observation:

We can see from the following graph of sales price vs. number of house, the target variable is **positively skewed** with a skew of **1.8829**.Reason behind skewness is that the samples available are mostly on the **lower** side of the price.

Now a distribution that is symmetric or nearly so is often easier to handle and interpret than a skewed distribution. So if the dataset is skewed, then an ML model wouldn’t be able to do a good job of prediction. More specifically, a normal or Gaussian distribution is often regarded as ideal as it is assumed by many statistical methods.

To reduce right or positive skewness, some approaches are taking-

1. **roots** or
2. **logarithms** or
3. **reciprocals**
   [1] #### Note: To reduce left skewness,we could take squares or cubes or higher powers.[1]

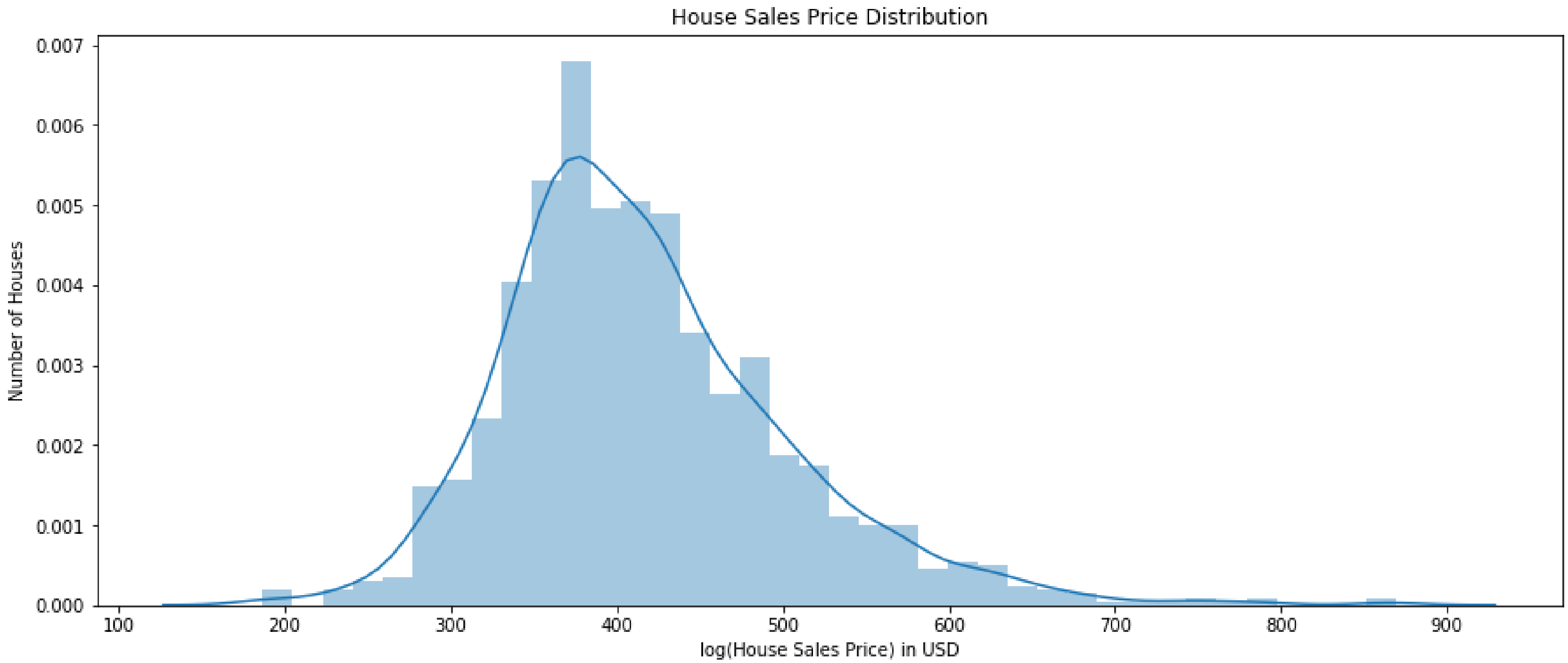
In [4]:
```
#Processing
#distribution of the target variable after taking square root

plt.figure(figsize=(15,6))
sns.distplot((np.sqrt(train.SalePrice)))
plt.xlabel("log(House Sales Price) in USD")
plt.ylabel("Number of Houses")
plt.title("House Sales Price Distribution")

print("Skew is:", (np.sqrt(train.SalePrice)).skew())
```

```
c:\python35\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multid
imensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be in
terpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different resul
t.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Skew is: 0.9431527373310963
```



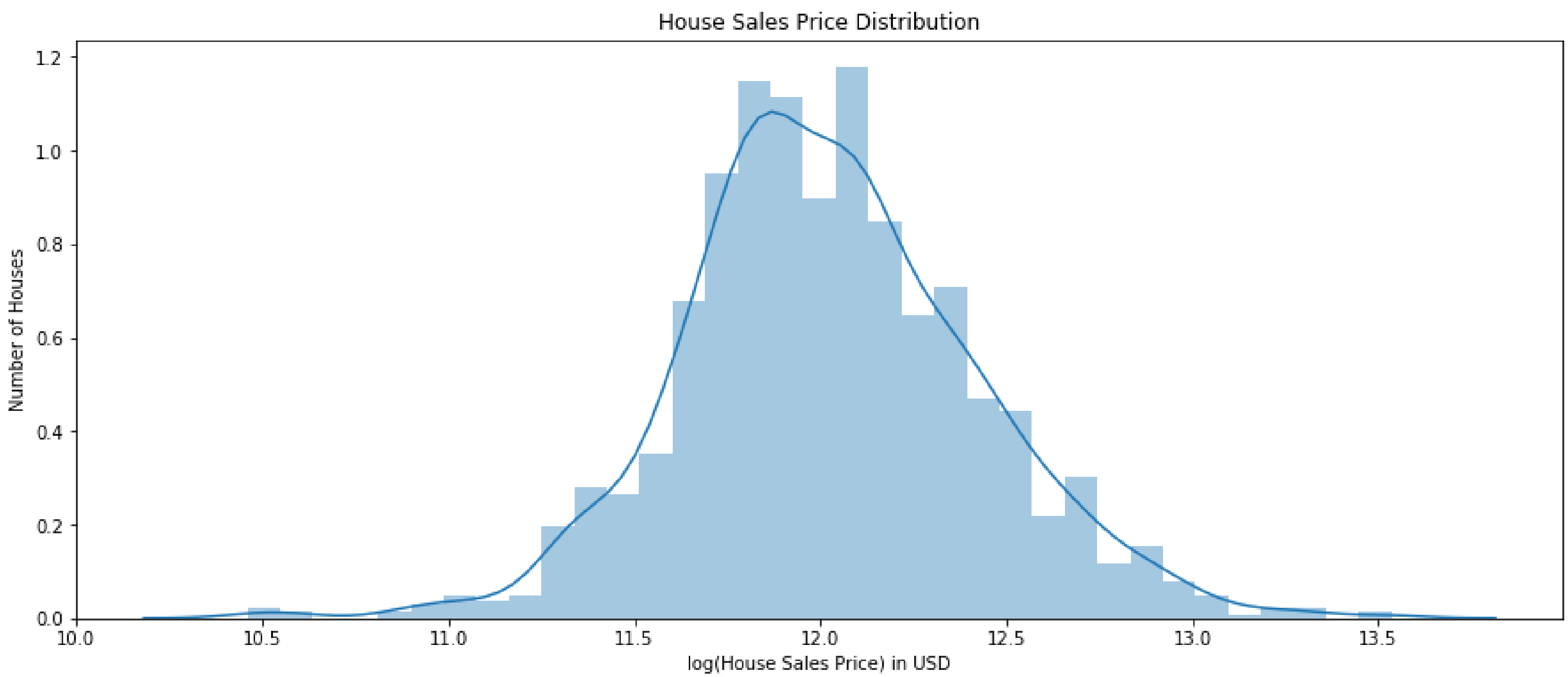
In [5]:
```
#Processing
#distribution of the target variable after taking log

plt.figure(figsize=(15,6))
sns.distplot((np.log(train.SalePrice)))
plt.xlabel("log(House Sales Price) in USD")
plt.ylabel("Number of Houses")
plt.title("House Sales Price Distribution")

print("Skew is:", (np.log(train.SalePrice)).skew())
```

```
c:\python35\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multid
imensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be in
terpreted as an array index, `arr[np.array(seq)]`, which will result either in an error or a different resul
t.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval

Skew is: 0.12133506220520406
```



2.1.2. Trials:

If we take the **square root** of the target variable, the skewness is reduced to **0.9431**. And if we take **logarithm** of the target variable, the skewness is **0.1213** and the target variable now almost follows a Gaussian distribution.

2.1.3. Action:

Since taking the logarithm of sales price reduces the skewness the most, the decision is to turn our target variable from **sales price** to **log (Sales price)**. Also we have to calculate RMSE value from log of predicted and log of observed value to implement this.

In [6]:
```
#processing
#Converting SalePrice to log value
train.SalePrice = np.log(train.SalePrice)
```

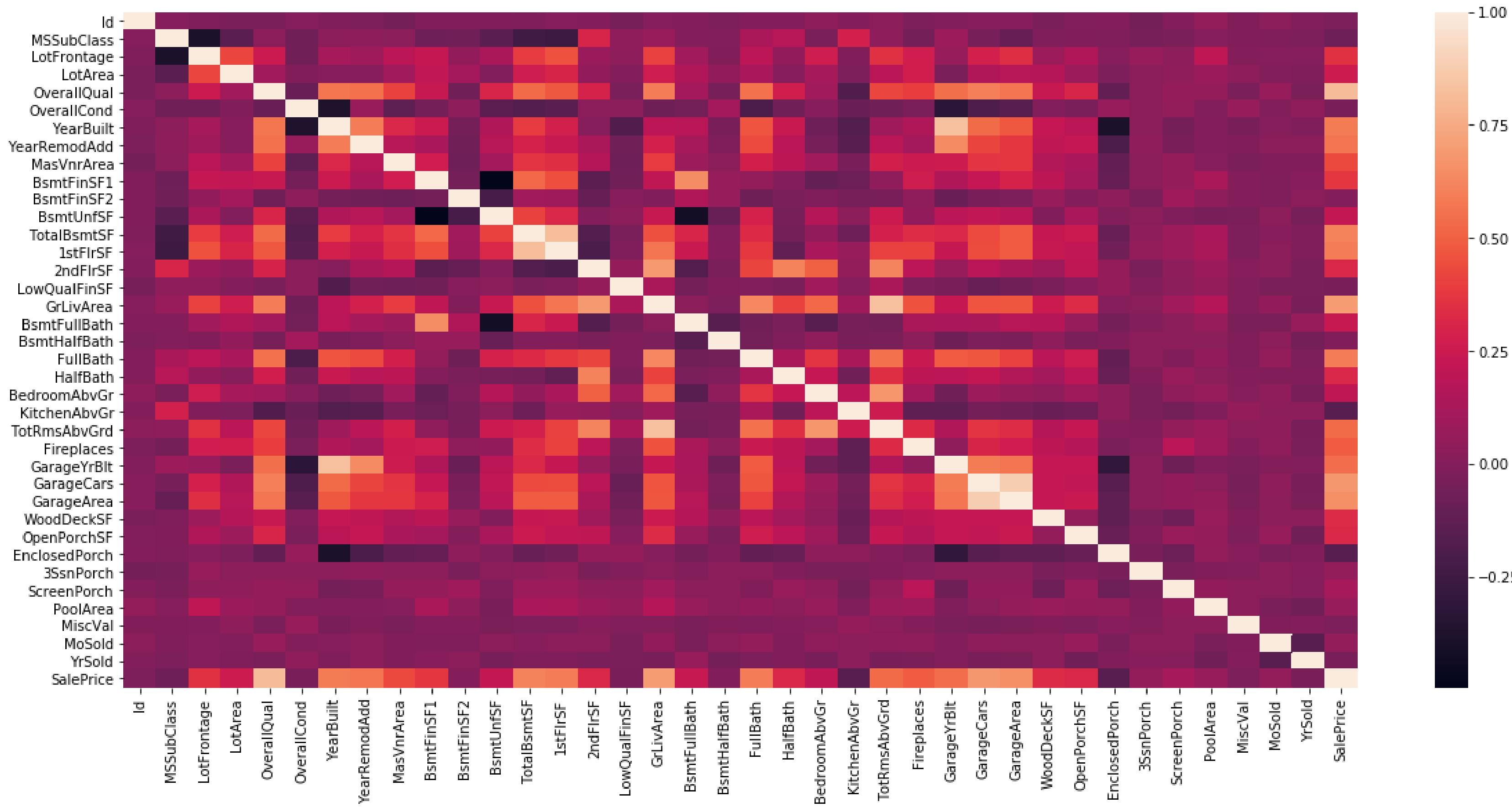
2.2. Numeric Features' Correlation Analysis:

The numeric features are usually correlated among them and with the target variable. Correlation measures a linear relation (or lack of it) such that one of the variables increases when the other one increases (positive correlation), or one of the variables increases when the other one decreases (negative correlation). So usually correlation cannot be measured for categorical variable.

In [7]:
```
#Extracting numeric features
numeric_features = train.select_dtypes(exclude='object')
```

In [8]:
```
#correlation matrix analysis
corr_df = numeric_features.corr()
f, ax = plt.subplots(figsize=(20, 9))
sns.heatmap(corr_df)
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x1fafc9fb6a0>



2.2.1. Observation:

From the correlation data frame and the heat map we can easily identify the features that are highly correlated between them and the target variables. We can see from the heat map that the diagonal has the highest value(1) of correlation which are actually correlation with its own value.

2.2.2. Action:

To exclude these values we can replace these values with a much lower value.Then we can find the features that are higly correlated.

In [9]:
```
#Finding features that are highly correlated with each other
np.fill_diagonal(corr_df.values,.001)
c1 = corr_df.abs().unstack()
c1.sort_values(ascending = False)
```

Out[9]:
```
GarageArea    GarageCars      0.882475
GarageCars    GarageArea      0.882475
GarageYrBlt   YearBuilt       0.825667
YearBuilt     GarageYrBlt     0.825667
GrLivArea     TotRmsAbvGrd    0.825489
TotRmsAbvGrd  GrLivArea       0.825489
1stFlrSF      TotalBsmtSF     0.819530
TotalBsmtSF   1stFlrSF        0.819530
SalePrice     OverallQual     0.817184
OverallQual   SalePrice       0.817184
SalePrice     GrLivArea       0.700927
GrLivArea     SalePrice       0.700927
              2ndFlrSF        0.687501
2ndFlrSF      GrLivArea       0.687501
SalePrice     GarageCars      0.680625
GarageCars    SalePrice       0.680625
BedroomAbvGr  TotRmsAbvGrd    0.676620
TotRmsAbvGrd  BedroomAbvGr    0.676620
SalePrice     GarageArea      0.650888
GarageArea    SalePrice       0.650888
BsmtFullBath  BsmtFinSF1      0.649212
BsmtFinSF1    BsmtFullBath    0.649212
YearRemodAdd  GarageYrBlt     0.642277
GarageYrBlt   YearRemodAdd    0.642277
GrLivArea     FullBath        0.630012
FullBath      GrLivArea       0.630012
TotRmsAbvGrd  2ndFlrSF        0.616423
2ndFlrSF      TotRmsAbvGrd    0.616423
TotalBsmtSF   SalePrice       0.612134
SalePrice     TotalBsmtSF     0.612134
                                ...
FullBath      FullBath        0.001000
TotalBsmtSF   TotalBsmtSF     0.001000
1stFlrSF      1stFlrSF        0.001000
OverallQual   OverallQual     0.001000
OverallCond   OverallCond     0.001000
YearBuilt     YearBuilt       0.001000
YearRemodAdd  YearRemodAdd    0.001000
MasVnrArea    MasVnrArea      0.001000
BsmtFinSF1    BsmtFinSF1      0.001000
BsmtFinSF2    BsmtFinSF2      0.001000
BsmtUnfSF     BsmtUnfSF       0.001000
Id            Id              0.001000
```

```
BsmtFullBath  BsmtFullBath    0.001000
2ndFlrSF      2ndFlrSF        0.001000
LowQualFinSF  LowQualFinSF    0.001000
GrLivArea     GrLivArea       0.001000
Id            YrSold          0.000712
YrSold        Id              0.000712
FullBath      LowQualFinSF    0.000710
LowQualFinSF  FullBath        0.000710
Id            OpenPorchSF     0.000477
OpenPorchSF   Id              0.000477
MiscVal       3SsnPorch       0.000354
3SsnPorch     MiscVal         0.000354
TotalBsmtSF   BsmtHalfBath    0.000315
BsmtHalfBath  TotalBsmtSF     0.000315
BsmtFullBath  3SsnPorch       0.000106
3SsnPorch     BsmtFullBath    0.000106
GarageYrBlt   Id              0.000072
Id            GarageYrBlt     0.000072
Length: 1444, dtype: float64
```

2.2.3. Observation:

Now there are **4 pairs** whose correlation values are greater than **.8** which means the two variables in each of these pairs have a highly linear relationship. The pairs are-
(a) GarageArea and GarageCars: 0.882475
(b) YearBuilt and GarageYrBlt: 0.82566
(c) GrLivArea and TotRmsAbvGrd: 0.825489
(d) 1stFlrSF and TotalBsmtSF: 0.819530

2.2.4. Analysis:

Among these, we need to find which pairs containing attributes of almost same physical significance. We can safely deduct that **higher garage area** ensures **more car space**, **higher ground living area** means there will be higher number of **total rooms above ground**. And finally **first floor area** must increase if **total basement area**y increases.

In [10]:
```
#Finding correlations of features with target attribute
print(corr_df['SalePrice'].sort_values(ascending=False)[:30], '\n')
print(corr_df['SalePrice'].sort_values(ascending=False)[-15:])
```

```
OverallQual     0.817184
GrLivArea       0.700927
GarageCars      0.680625
GarageArea      0.650888
TotalBsmtSF     0.612134
1stFlrSF        0.596981
FullBath        0.594771
YearBuilt       0.586570
YearRemodAdd    0.565608
GarageYrBlt     0.541073
TotRmsAbvGrd    0.534422
Fireplaces      0.489449
MasVnrArea      0.430809
BsmtFinSF1      0.372023
LotFrontage     0.355878
WoodDeckSF      0.334135
OpenPorchSF     0.321053
2ndFlrSF        0.319300
HalfBath        0.313982
LotArea         0.257320
BsmtFullBath    0.236224
BsmtUnfSF       0.221985
BedroomAbvGr    0.209044
ScreenPorch     0.121208
PoolArea        0.069798
MoSold          0.057329
3SsnPorch       0.054900
BsmtFinSF2      0.004832
SalePrice       0.001000
BsmtHalfBath   -0.005149
Name: SalePrice, dtype: float64

ScreenPorch      0.121208
PoolArea         0.069798
MoSold           0.057329
3SsnPorch        0.054900
BsmtFinSF2       0.004832
SalePrice        0.001000
BsmtHalfBath    -0.005149
Id              -0.017942
MiscVal         -0.020021
OverallCond     -0.036868
YrSold          -0.037263
```

```
LowQualFinSF    -0.037963
MSSubClass      -0.073959
KitchenAbvGr    -0.147548
EnclosedPorch   -0.149050
Name: SalePrice, dtype: float64
```

2.2.5. Decision of Action:

After finding out the pairs that carry almost same attributes in sense of physical significance, we can safely **drop** one from each pair having **lower** correlation with **target variable**. They are-
1.GarageArea
2.TotRmsAbvGrd
3.1stFrSF

We will drop these features in data processing steps and observe the change in error accordingly.

Note:

Since **Year built and garage year built** have different physical significance but high correlation,we will deal with this feature later.

2.3. Outlier handling:

Outliers: In statistics, an outlier is an observation point that is distant from other observations.[2]
Causes:[2]

1. Mistake in data collection
2. Indication of variance in data

2.3.1. Ways of Finding outlier:[3]

Univariate:

1. Boxplot

Multivariate:

1. Scatterplolt
2. Z score
3. IQR Score

2.3.2. Processing options:

1. Deleting samples that include outlier
   -This is only applicable for train data. Beacuse kaggle demands use of all test data during predictions.
2. Replacing by closest non-outlier
3. Replacing by mean (Showed with result)[5]
4. If data seems to be a mistake, then can be replaced by 'Missing'
5. Using the concept of KNN to replace the outlier

2.4.Missing value handling:

Missing values can mean different things. It can be either be truly missing or it can mean the sample does not have that feature.

2.4.1. Processing options:
[4]

1. Delete the sample if most attributes are missing(only apllicable train data)
2. Using the concept of K nearest neighbour(if the value is missed somehow during data collection)
3. Using the most common value of that feature(if the feature is a must for a house))
4. Replacing with zero(if the sample does not have that particular feature)

2.5. Skewed features handling:

As mentioned before, unskewed data tends to give better results. So all the numeric features that has high skewness needs processing so that their skewness is reduced.

2.5.1. Observation and action:

After trials it was seen that features having skewness over 0.35 needs to reduce their skewness.So the limit set for reducing skewness was set to 0.35.

2.6. Feature adding:

- Since a new house should have a higher price than an old house, a new binary feature has been added where 1 means the house is new and 0 means the house is old. If the Year sold is greater that Year built than the house is old, and if theye are equal the house is new.
- Same assumptions were made for a house that has been remodeled.
- Overall area should be a important factor indeciding the price of the house. So a new feature has been added named 'OverallSF'which adds the the 2nd floor area and the total basement area.[5]

2.7. Categorical features handling:

Categorical features have been trasformed to numeric features by using one hot encoding. Because our models accept numeric features only.

2.8. Scaling the features:

As the values of different attributes are at different scales, they are needed to be scaled.
Procedures for scaling:

1. Min Max Scaler
2. Robust Scaler
3. Standard Scaler
   Robust scaler gave the best performance beacuse it works better with outliers inside data and our data still includes some outliers.

3.Preprocessing steps:

Before applying any action, we need to merge the dataset so that we do not need to perform the cleaning and processing steps twice.

3.1. Merging dataframe

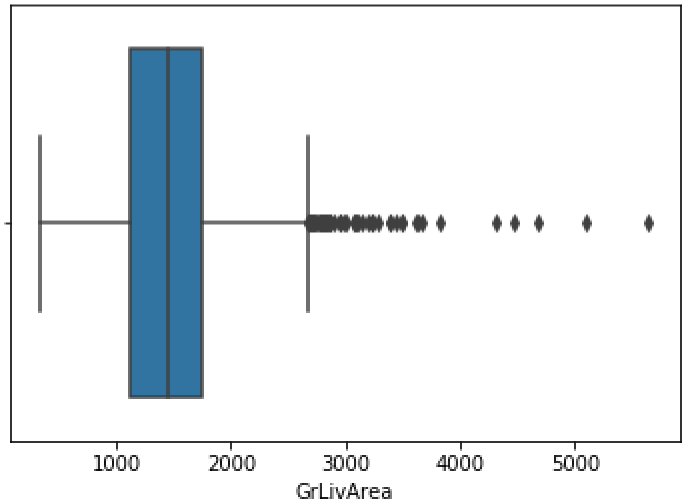
In [11]:
```
train['source']='train'
test['source']='test'
data = pd.concat([train,test], ignore_index = True, sort = False)
print(train.shape, test.shape, data.shape)
```

```
(1460, 82) (1459, 81) (2919, 82)
```

3.2. Outlier finding

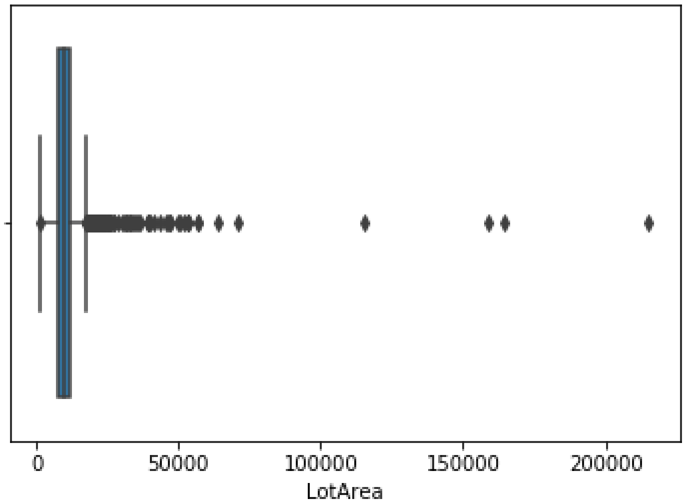
In [12]:
```
sns.boxplot(x=data['GrLivArea'])
```

Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x1fafcb895c0>



In [13]:
```
sns.boxplot(x= data['LotArea'])
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1fafcbcf710>



3.3. Outlier reduction

Outliers for 'GrLivArea' and 'LotArea' were dropped. Drooping other outliers did not improve results satisfactorily.

In [14]:
```
GrLivArea_mean = data['GrLivArea'].mean()
func = lambda x: x['GrLivArea'] > 4000 and GrLivArea_mean or x['GrLivArea']
data['GrLivArea'] = data.apply(func,axis=1).astype(float)

LotArea_mean = data['LotArea'].mean()
func = lambda x: x['LotArea'] > 50000 and LotArea_mean or x['LotArea']
data['LotArea'] = data.apply(func,axis=1).astype(float)
```

3.4. Removing features

In [15]:
```
#Removing unnecessary features(decided from correlation values)

del data['Id']
del data['SalePrice']
del data['GarageArea']
del data['1stFlrSF']
del data['TotRmsAbvGrd']
del data['BedroomAbvGr']
```

3.5. Missing value handling

3.5.1 Missing value handling: Categorical features not present

In [16]:
```
# Missing value of Categorical Features
# Missing values that are present because the houses does not have these features
cat_features_not_present = data[['Alley','BsmtCond','BsmtQual','BsmtExposure','BsmtFinType1', 'BsmtFinType2',
'FireplaceQu','GarageType','GarageFinish', 'GarageQual', 'GarageCond','PoolQC','Fence','MiscFeature']]
```

In [17]:
```
for i in list(cat_features_not_present.columns.values):
    print("\n")
    print("Analysing the " + i)
    print(cat_features_not_present[i].value_counts())
```

```
Analysing the Alley
Grvl    120
Pave     78
Name: Alley, dtype: int64


Analysing the BsmtCond
TA    2606
Gd     122
Fa     104
Po       5
Name: BsmtCond, dtype: int64


Analysing the BsmtQual
TA    1283
Gd    1209
Ex     258
Fa      88
Name: BsmtQual, dtype: int64


Analysing the BsmtExposure
No    1904
Av     418
Gd     276
Mn     239
Name: BsmtExposure, dtype: int64


Analysing the BsmtFinType1
Unf    851
GLQ    849
ALQ    429
Rec    288
BLQ    269
LwQ    154
Name: BsmtFinType1, dtype: int64


Analysing the BsmtFinType2
Unf    2493
```

```
Rec     105
LwQ      87
BLQ      68
ALQ      52
GLQ      34
Name: BsmtFinType2, dtype: int64


Analysing the FireplaceQu
Gd    744
TA    592
Fa     74
Po     46
Ex     43
Name: FireplaceQu, dtype: int64


Analysing the GarageType
Attchd     1723
Detchd      779
BuiltIn     186
Basment      36
2Types       23
CarPort      15
Name: GarageType, dtype: int64


Analysing the GarageFinish
Unf    1230
RFn     811
Fin     719
Name: GarageFinish, dtype: int64


Analysing the GarageQual
TA    2604
Fa     124
Gd      24
Po       5
Ex       3
Name: GarageQual, dtype: int64
```

```
Analysing the GarageCond
TA    2654
Fa      74
Gd      15
Po      14
Ex       3
Name: GarageCond, dtype: int64


Analysing the PoolQC
Ex    4
Gd    4
Fa    2
Name: PoolQC, dtype: int64


Analysing the Fence
MnPrv    329
GdPrv    118
GdWo     112
MnWw      12
Name: Fence, dtype: int64


Analysing the MiscFeature
Shed    95
Gar2     5
Othr     4
TenC     1
Name: MiscFeature, dtype: int64
```

In [18]:

```
#Replacing values with a new class "None"
data.update(cat_features_not_present.fillna("None"))
```

3.5.2. Missing value handling: Numeric features not present

In [19]:
```
#missing values of numerical features
# Missing values that are present because the houses does not have these features
num_feat_not_present= data[['BsmtFullBath','BsmtHalfBath', 'TotalBsmtSF','BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnf
SF','GarageCars']]
```

In [20]:
```
for i in list(num_feat_not_present.columns.values):
    print("")
    print("Analysing the " + i)
    print(num_feat_not_present[i].value_counts())
```

```
Analysing the BsmtFullBath
0.0    1705
1.0    1172
2.0      38
3.0       2
Name: BsmtFullBath, dtype: int64

Analysing the BsmtHalfBath
0.0    2742
1.0     171
2.0       4
Name: BsmtHalfBath, dtype: int64

Analysing the TotalBsmtSF
0.0       78
864.0     74
672.0     29
912.0     26
1040.0    25
768.0     24
816.0     23
728.0     20
1008.0    19
780.0     19
384.0     19
960.0     18
894.0     17
756.0     17
832.0     17
546.0     16
936.0     16
720.0     16
600.0     16
848.0     16
483.0     14
630.0     13
952.0     13
840.0     13
988.0     12
624.0     12
876.0     11
784.0     11
796.0     11
```

```
1056.0    10
          ..
1967.0     1
1905.0     1
1679.0     1
1533.0     1
2140.0     1
1994.0     1
1378.0     1
763.0      1
1047.0     1
1376.0     1
904.0      1
370.0      1
1570.0     1
2033.0     1
1709.0     1
1519.0     1
2077.0     1
1550.0     1
797.0      1
699.0      1
559.0      1
396.0      1
1866.0     1
1641.0     1
961.0      1
1949.0     1
1231.0     1
1829.0     1
1475.0     1
1243.0     1
Name: TotalBsmtSF, Length: 1058, dtype: int64

Analysing the BsmtFinSF1
0.0       929
24.0       27
16.0       14
300.0       9
288.0       8
384.0       8
600.0       8
20.0        8
```

```
602.0      7
500.0      7
700.0      7
360.0      7
456.0      7
936.0      7
375.0      7
624.0      7
560.0      6
312.0      6
528.0      6
504.0      6
662.0      6
547.0      6
468.0      6
368.0      6
544.0      6
553.0      6
120.0      6
276.0      6
625.0      6
588.0      6
          ..
1285.0     1
1150.0     1
806.0      1
349.0      1
1682.0     1
702.0      1
393.0      1
587.0      1
427.0      1
586.0      1
1836.0     1
501.0      1
954.0      1
710.0      1
722.0      1
491.0      1
1812.0     1
1261.0     1
1375.0     1
1172.0     1
```

```
987.0       1
759.0       1
1178.0      1
1158.0      1
1122.0      1
1022.0      1
939.0       1
1124.0      1
1619.0      1
1106.0      1
Name: BsmtFinSF1, Length: 991, dtype: int64

Analysing the BsmtFinSF2
0.0       2571
294.0        5
180.0        5
162.0        3
539.0        3
168.0        3
147.0        3
144.0        3
483.0        3
374.0        3
435.0        3
182.0        2
174.0        2
210.0        2
202.0        2
273.0        2
117.0        2
712.0        2
288.0        2
41.0         2
105.0        2
116.0        2
344.0        2
596.0        2
110.0        2
492.0        2
72.0         2
465.0        2
469.0        2
252.0        2
```

```
          ...
63.0        1
278.0       1
530.0       1
402.0       1
286.0       1
884.0       1
163.0       1
177.0       1
334.0       1
532.0       1
258.0       1
215.0       1
690.0       1
506.0       1
1085.0      1
263.0       1
404.0       1
411.0       1
981.0       1
691.0       1
713.0       1
912.0       1
156.0       1
66.0        1
488.0       1
196.0       1
904.0       1
456.0       1
624.0       1
823.0       1
Name: BsmtFinSF2, Length: 272, dtype: int64

Analysing the BsmtUnfSF
0.0       241
384.0      19
728.0      14
672.0      13
600.0      12
572.0      11
216.0      11
100.0      11
816.0      11
```

```
624.0      10
270.0      10
300.0      10
264.0       9
396.0       9
280.0       9
186.0       9
768.0       9
780.0       8
546.0       8
348.0       8
294.0       8
440.0       8
162.0       8
480.0       8
832.0       8
108.0       8
840.0       8
784.0       7
80.0        7
392.0       7
           ..
127.0       1
795.0       1
214.0       1
1098.0      1
584.0       1
532.0       1
983.0       1
79.0        1
388.0       1
559.0       1
1616.0      1
889.0       1
1078.0      1
1411.0      1
999.0       1
659.0       1
709.0       1
1214.0      1
657.0       1
587.0       1
1146.0      1
```

```
2140.0      1
579.0       1
735.0       1
1073.0      1
1503.0      1
445.0       1
958.0       1
1559.0      1
1369.0      1
Name: BsmtUnfSF, Length: 1135, dtype: int64

Analysing the GarageCars
2.0    1594
1.0     776
3.0     374
0.0     157
4.0      16
5.0       1
Name: GarageCars, dtype: int64
```

In [21]:
```
#Replacing values with value '0.0'
data.update(num_feat_not_present.fillna(0.0))
```

3.5.3. Missing value handling: Time related feature with high correlation

In [22]:
```
#Exceptional feature: Garage year built: If missing then replace by year built. Because most likely it was bu
ilt with the house
data.update(data['GarageYrBlt'].fillna(data.YearBuilt))
```

3.5.3.4. Random Missing value handling:

In [23]:
```
#Missing values at random
#Features that are must for a house
#Replaced by most common values
catfeats_fillnamode = ['Electrical', 'MasVnrType', 'MSZoning', 'Functional', 'Utilities','Exterior1st', 'Exterior2nd', 'KitchenQual', 'SaleType']
data[catfeats_fillnamode].fillna(data[catfeats_fillnamode].mode().iloc[0])
```

Out[23]:

	Electrical	MasVnrType	MSZoning	Functional	Utilities	Exterior1st	Exterior2nd	KitchenQual	SaleType
0	SBrkr	BrkFace	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
1	SBrkr	None	RL	Typ	AllPub	MetalSd	MetalSd	TA	WD
2	SBrkr	BrkFace	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
3	SBrkr	None	RL	Typ	AllPub	Wd Sdng	Wd Shng	Gd	WD
4	SBrkr	BrkFace	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
5	SBrkr	None	RL	Typ	AllPub	VinylSd	VinylSd	TA	WD
6	SBrkr	Stone	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
7	SBrkr	Stone	RL	Typ	AllPub	HdBoard	HdBoard	TA	WD
8	FuseF	None	RM	Min1	AllPub	BrkFace	Wd Shng	TA	WD
9	SBrkr	None	RL	Typ	AllPub	MetalSd	MetalSd	TA	WD
10	SBrkr	None	RL	Typ	AllPub	HdBoard	HdBoard	TA	WD
11	SBrkr	Stone	RL	Typ	AllPub	WdShing	Wd Shng	Ex	New
12	SBrkr	None	RL	Typ	AllPub	HdBoard	Plywood	TA	WD
13	SBrkr	Stone	RL	Typ	AllPub	VinylSd	VinylSd	Gd	New
14	SBrkr	BrkFace	RL	Typ	AllPub	MetalSd	MetalSd	TA	WD
15	FuseA	None	RM	Typ	AllPub	Wd Sdng	Wd Sdng	TA	WD
16	SBrkr	BrkFace	RL	Typ	AllPub	Wd Sdng	Wd Sdng	TA	WD
17	SBrkr	None	RL	Typ	AllPub	MetalSd	MetalSd	TA	WD
18	SBrkr	None	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
19	SBrkr	None	RL	Min1	AllPub	BrkFace	Plywood	TA	COD
20	SBrkr	BrkFace	RL	Typ	AllPub	VinylSd	VinylSd	Gd	New
21	FuseF	None	RM	Typ	AllPub	Wd Sdng	Wd Sdng	Gd	WD
22	SBrkr	BrkFace	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
23	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
24	SBrkr	None	RL	Typ	AllPub	Plywood	Plywood	Gd	WD
25	SBrkr	Stone	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD

	Electrical	MasVnrType	MSZoning	Functional	Utilities	Exterior1st	Exterior2nd	KitchenQual	SaleType
26	SBrkr	None	RL	Typ	AllPub	Wd Sdng	Wd Sdng	Gd	WD
27	SBrkr	Stone	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
28	SBrkr	None	RL	Typ	AllPub	MetalSd	MetalSd	TA	WD
29	SBrkr	None	RM	Typ	AllPub	MetalSd	MetalSd	Fa	WD
...	...	...	...	...	...	...	...	...	...
2889	SBrkr	None	RM	Typ	AllPub	MetalSd	MetalSd	Fa	WD
2890	SBrkr	BrkFace	RM	Typ	AllPub	MetalSd	MetalSd	Gd	WD
2891	SBrkr	None	C (all)	Mod	AllPub	Wd Sdng	Wd Sdng	TA	WD
2892	SBrkr	None	C (all)	Min2	AllPub	WdShing	Wd Shng	TA	WD
2893	SBrkr	None	C (all)	Typ	AllPub	MetalSd	MetalSd	TA	WD
2894	SBrkr	Stone	RM	Typ	AllPub	CemntBd	CmentBd	Ex	New
2895	SBrkr	Stone	RM	Typ	AllPub	CemntBd	CmentBd	Gd	WD
2896	SBrkr	None	RL	Typ	AllPub	Plywood	Plywood	Fa	WD
2897	SBrkr	None	RL	Typ	AllPub	Plywood	Plywood	TA	WD
2898	SBrkr	None	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
2899	SBrkr	BrkFace	RL	Typ	AllPub	Plywood	Plywood	TA	WD
2900	SBrkr	None	RL	Typ	AllPub	Plywood	Plywood	TA	WD
2901	SBrkr	None	RL	Typ	AllPub	VinylSd	VinylSd	Gd	WD
2902	SBrkr	Stone	RL	Typ	AllPub	VinylSd	VinylSd	Gd	New
2903	SBrkr	BrkFace	RL	Typ	AllPub	VinylSd	VinylSd	Ex	New
2904	FuseA	None	RL	Mod	AllPub	CBlock	VinylSd	TA	WD
2905	SBrkr	BrkFace	RM	Typ	AllPub	MetalSd	MetalSd	TA	WD
2906	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
2907	SBrkr	None	RL	Typ	AllPub	Plywood	Plywood	TA	WD
2908	SBrkr	None	RL	Typ	AllPub	Plywood	Plywood	TA	WD
2909	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
2910	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD

	Electrical	MasVnrType	MSZoning	Functional	Utilities	Exterior1st	Exterior2nd	KitchenQual	SaleType
2911	SBrkr	BrkFace	RL	Typ	AllPub	Plywood	Plywood	TA	WD
2912	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
2913	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
2914	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
2915	SBrkr	None	RM	Typ	AllPub	CemntBd	CmentBd	TA	WD
2916	SBrkr	None	RL	Typ	AllPub	VinylSd	VinylSd	TA	WD
2917	SBrkr	None	RL	Typ	AllPub	HdBoard	Wd Shng	TA	WD
2918	SBrkr	BrkFace	RL	Typ	AllPub	HdBoard	HdBoard	TA	WD

2919 rows × 9 columns

3.6. Skewness handling:

In [24]:
```
#log transform skewed numeric features:
#Features having skew value over .75 are unskewed
numeric_feats = data.dtypes[data.dtypes != "object"].index
skewed_feats = data[numeric_feats].apply(lambda x: pd.DataFrame.skew(x)) #compute skewness
skewed_feats = skewed_feats[skewed_feats > 0.35]
skewed_feats = skewed_feats.index
data[skewed_feats] = np.log1p(data[skewed_feats])
```

3.7. Feature adding:

In [25]:
```
#Feature adding1
data['Remodeled'] = data['YearRemodAdd'] - data['YearBuilt']
func = lambda x: x['Remodeled'] > 0 and 1.0 or 0.0
data['Remodeled'] = data.apply(func,axis=1).astype(float)
```

In [26]:
```
#Feature adding 2
data['NewHouse'] = data['YrSold'] - data['YearBuilt']
func = lambda x: x['NewHouse'] == 0 and 1.0 or 0.0
data['NewHouse'] = data.apply(func,axis=1).astype(float)
```

In [27]:
```
#Feature adding 3
data['OverallSF'] = data['2ndFlrSF'] + data['TotalBsmtSF']
```

3.8. Preprocessing for other irrarional values

In [28]:
```
#Preprocessing for other irrational values
data.replace(np.nan,0,inplace=True)
data.replace(np.inf,0,inplace=True)
data = data.fillna(0)
```

3.9. One hot encoding and scaling

In [29]:
```
#One Hot Encoding and Scaling
from sklearn.preprocessing import RobustScaler
dm_data = pd.get_dummies(data)
robust_scaler = RobustScaler()
dm_data = robust_scaler.fit_transform(dm_data)
```

3.10 Preparations for running model

In [30]:
```
#Spliting merged data
ntrain = train.shape[0]
ntest = test.shape[0]
labels_train= train.SalePrice.values
train_df = pd.DataFrame(dm_data[:ntrain])
test_df = pd.DataFrame(dm_data[ntrain:])
```

In [31]:
```
#Steps for fast calculation
train_df =np.array(train_df,dtype = np.float64)
labels_train = np.array(labels_train,dtype = np.float64)
test_df = np.array(test_df,dtype = np.float64)
```

In [32]:
```
labels_train = labels_train.reshape(-1,1)
```

In [33]:
```
#final number of input features
train_df.shape[1]
```

Out[33]:
```
312
```

4. ANN model

4.1. Model building for Neural Network

4.1.1 Functions

In [34]:
```
def create_ann_model(LearningRate,epoch,reg_const):
    input_node = train_df.shape[1]
    hlayer_1 = input_node/2
    hlayer_2 = 2
    output_node = 1
    with tf.variable_scope("WB", reuse=tf.AUTO_REUSE):
        w1= tf.Variable(tf.get_variable('w1',[input_node, hlayer_1],dtype=tf.float64))
        w2= tf.Variable(tf.get_variable('w2',[hlayer_1, hlayer_2],dtype=tf.float64))
        w3= tf.Variable(tf.get_variable('w3',[hlayer_2, output_node],dtype=tf.float64))
    b1= tf.Variable(tf.zeros([hlayer_1],dtype=tf.float64))
    b2= tf.Variable(tf.zeros([hlayer_2],dtype=tf.float64))
    b3= tf.Variable(tf.zeros([output_node],dtype=tf.float64))

    weights= [w1,w2,w3]
    biases = [b1,b2,b3]

    input_feat = tf.placeholder(tf.float64, shape=[None, train_df.shape[1]])
    output_labels=tf.placeholder(tf.float64, shape=[None, 1])
    hl1 = tf.add(tf.matmul(input_feat, weights[0]),biases[0])
    hl2 = tf.add(tf.matmul(hl1, weights[1]),biases[1])
    out = tf.add(tf.matmul(hl2, weights[2]),biases[2])
    mse_loss = tf.losses.mean_squared_error(labels=output_labels, predictions=out)
    mse_loss=tf.cast(mse_loss,tf.float64)

    reg_const=np.float64(reg_const)
    _reg = 0
    for wt in weights:
        _reg=_reg+ 0.5*tf.reduce_sum(tf.square(wt))   #Applying L2 regularization formula
    reg=tf.cast(_reg,tf.float64)
    reg_loss =tf.multiply(np.float64(0.5),tf.add(mse_loss,tf.multiply(reg_const,reg)))
    optimizer = tf.train.RMSPropOptimizer(LearningRate)
    func_to_opt = optimizer.minimize(reg_loss)

    return mse_loss,out,func_to_opt,input_feat,output_labels
```

In [35]:
```
def train_loop(LearningRate,epoch,reg_const,train_data,train_lb,val_data,val_lb):
    loss_tr=[]
    loss_va=[]
    mse_loss,out,func_to_opt,input_feat,output_labels=create_ann_model(LearningRate,epoch,reg_const)

    sess = tf.Session()
    init = tf.global_variables_initializer()
    sess.run(init)


    for i in range(epoch):

        train_loss=0
        for j in range(0,len(train_data)):
            single_train_batch={input_feat:train_data[j],output_labels:train_lb[j]}
            _= sess.run((func_to_opt),feed_dict=single_train_batch)
            trainbatch_loss = sess.run((mse_loss),feed_dict=single_train_batch)
            train_loss+=trainbatch_loss
        train_loss=train_loss/(j+1)

        val_loss=0
        for k in range(len(val_data)):
            single_cv_batch={input_feat:val_data[k],output_labels:val_lb[k]}
            valid_loss = sess.run((mse_loss),feed_dict=single_cv_batch)
            val_loss+=valid_loss

        val_loss=val_loss/(k+1)
        loss_tr.append(train_loss)
        loss_va.append(val_loss)

        print('epoch:',i,'train_loss',train_loss,'valid_loss',val_loss)

    return input_feat,sess,out,loss_tr,loss_va
```

4.1.2. Grid search

Grid search has been done and loss values from the dictionary has been plotted to find the best model.A demo of the grid search process has been showed below.

In [36]:
```
no_of_batch=80
no_of_train_batch=int(0.7*len(train_df))
train_data=np.array_split(train_df[:no_of_train_batch],no_of_batch)
train_lb=np.array_split(labels_train[:no_of_train_batch],no_of_batch)
val_data=np.array_split(train_df[no_of_train_batch:],no_of_batch)
val_lb=np.array_split(labels_train[no_of_train_batch:],no_of_batch)

#epoch_list=[200,400,600]
epoch_val = 50
#LearningRates_list =[0.0001,0.001,0.1]
LearningRate_val = .0001
reg_const_list = [0.9,0.1,0.01,.001]
error_dict = {}
for reg_const_val in reg_const_list:
    input_feat,sess,out,loss_tr_gr,loss_val_gr=train_loop(LearningRate_val,epoch_val,reg_const_val,train_data
,train_lb,val_data,val_lb)
    error_dict['lr-'+str(reg_const_val)]=[loss_tr_gr,loss_val_gr]
```

Out[36]:
```
"no_of_batch=80\nno_of_train_batch=int(0.7*len(train_df))\ntrain_data=np.array_split(train_df[:no_of_train_ba
tch],no_of_batch)\ntrain_lb=np.array_split(labels_train[:no_of_train_batch],no_of_batch)\nval_data=np.array_s
plit(train_df[no_of_train_batch:],no_of_batch)\nval_lb=np.array_split(labels_train[no_of_train_batch:],no_of_
batch)\n\n#epoch_list=[200,400,600]\nepoch_val = 50\n#LearningRates_list =[0.0001,0.001,0.1]\nLearningRate_va
l = .0001\nreg_const_list = [0.9,0.1,0.01,.001]\nerror_dict = {}\nfor reg_const_val in reg_const_list:\n    i
nput_feat,sess,out,loss_tr_gr,loss_val_gr=train_loop(LearningRate_val,epoch_val,reg_const_val,train_data,trai
n_lb,val_data,val_lb)\n    error_dict['lr-'+str(reg_const_val)]=[loss_tr_gr,loss_val_gr]"
```

In [37]:
```
print(error_dict)
```

4.2.Highlights from hypermarameter tuning process

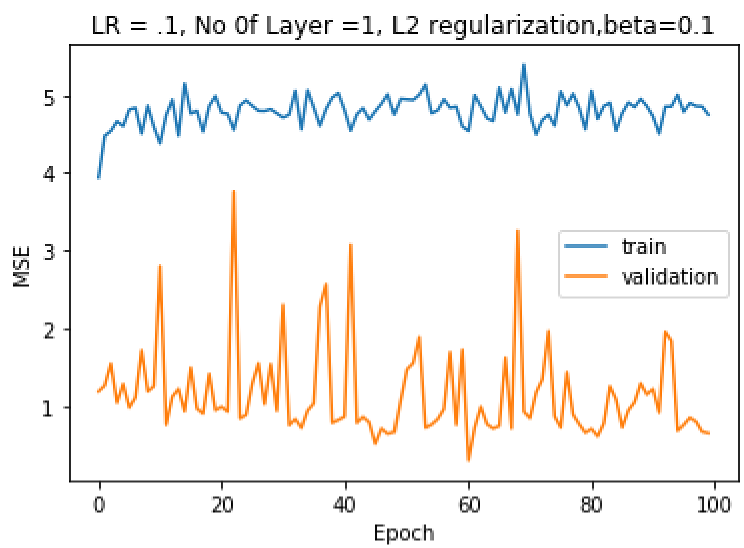
Note: The tuning process showed in the one layer model was followed for all the other models. Grid search and cross valudation were used to find the best tuned hyperparameters. Only the mentionable results are presented here to describe the process properly.

4.2.1. Layer:1(No hidden layer)

4.2.1.1. Learning rate tuning

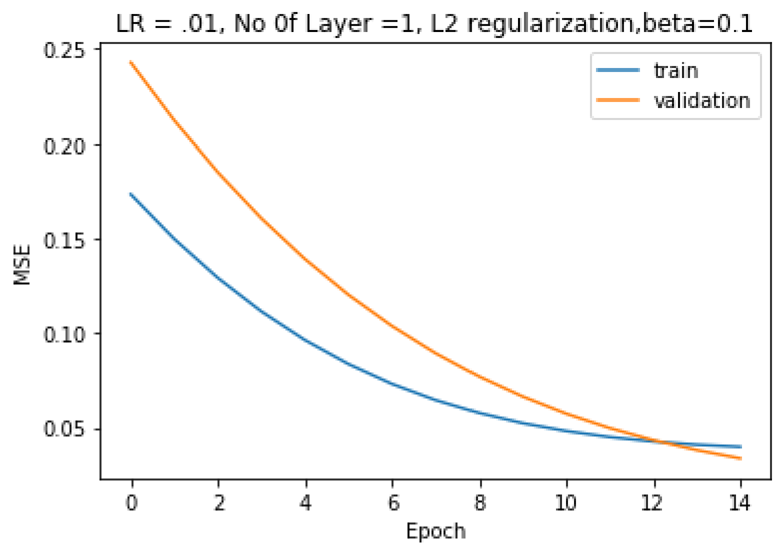
In a no hidden layer NN model,various learning rates were used for tuning.

1. At LearningRate = .1:
   Epoch value was 100.The learning rate is too high so the loss fluctuates from the very beginning.



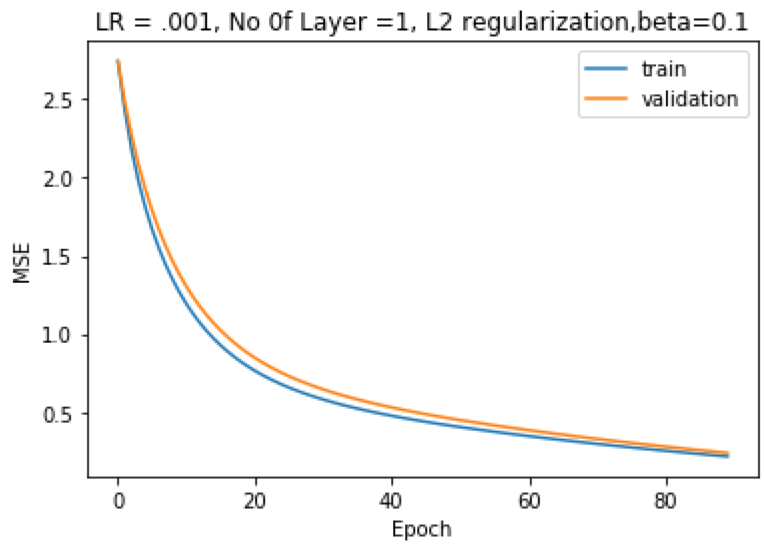
1. At LearningRate = .01:
   Here number of epoch was 100.After about 25 epochs, the learning rate started increasing slightly while the validation loss was decreasing.This can mean the learning rate is still high.The graph shown here starts from eporch value of 10.

```
plt.title('LR = .01, No 0f Layer =1, L2 regularization,beta=0.1')
plt.plot(loss_tr[10:], label = 'train')
plt.plot(loss_va[10:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



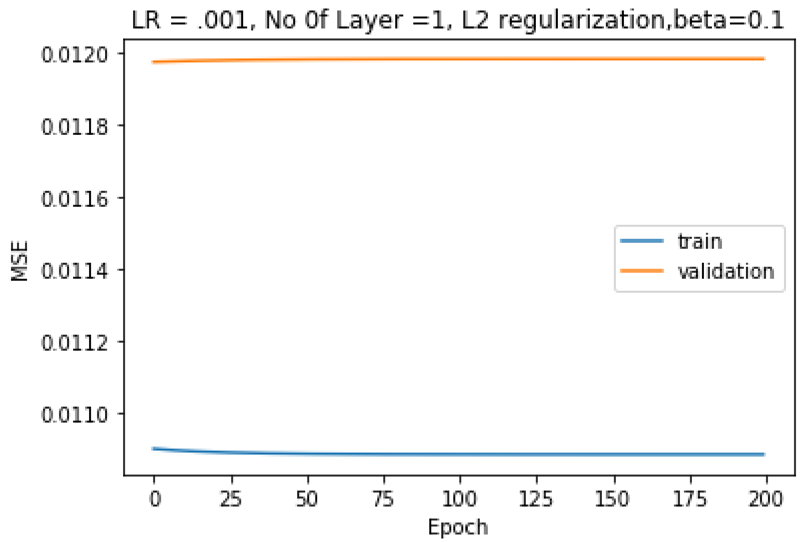
1. At LearningRate = .001:
   Epoch=100(Showed from 10)
   Training RMSE = .466
   Validation RMSE = .489
   Since the loss is still high and both training and validation losses are decreasing, epochs must be increased in next trial.

```
plt.title('LR = .001, No Of Layer =1, L2 regularization,beta=0.1')
plt.plot(loss_tr[10:], label = 'train')
plt.plot(loss_va[10:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



1. At Learning rate= .001 and Epoch = 500:
   Training RMSE = .10431
   Validation RMSE = .10946
   Here epoch values are shown from 300 and we can see that the error almost remains constant from 300 to 500 epochs.So epoch values above 300 will be unnecessary.

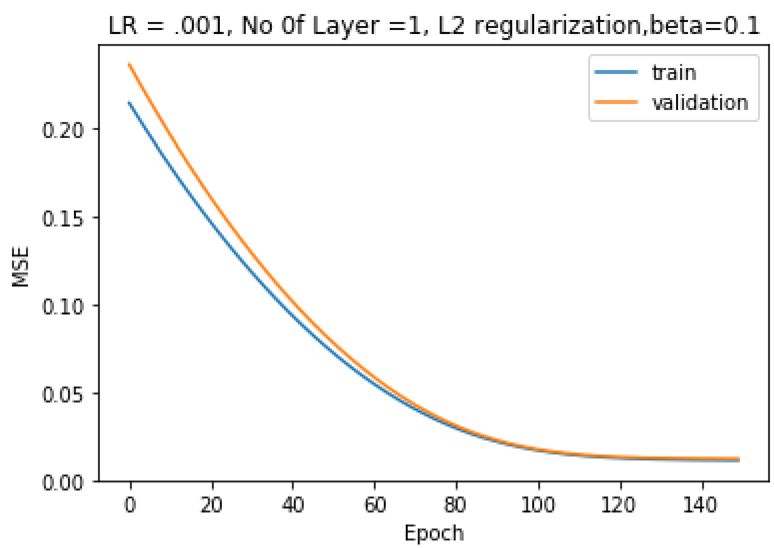
```
plt.title('LR = .001, No 0f Layer =1, L2 regularization,beta=0.1')
plt.plot(loss_tr[300:], label = 'train')
plt.plot(loss_va[300:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



Best result of Layer 1

1. At Learning Rate = .001 and Epoch = 250:
   Training RMSE: .105
   Validation RMSE: .1096
   (Epoch showed from 100)

```
plt.title('LR = .001, No 0f Layer =1, L2 regularization,beta=0.1')
plt.plot(loss_tr[100:], label = 'train')
plt.plot(loss_va[100:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

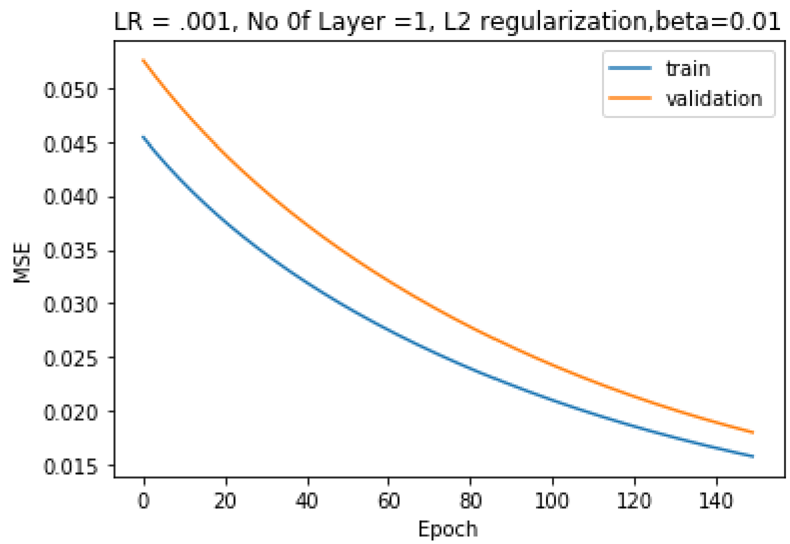


4.2.1.2. Regularization constant tuning:

Best value was found to be .1 for 1 layer model. Optimum values for other models were tuned accordingly.

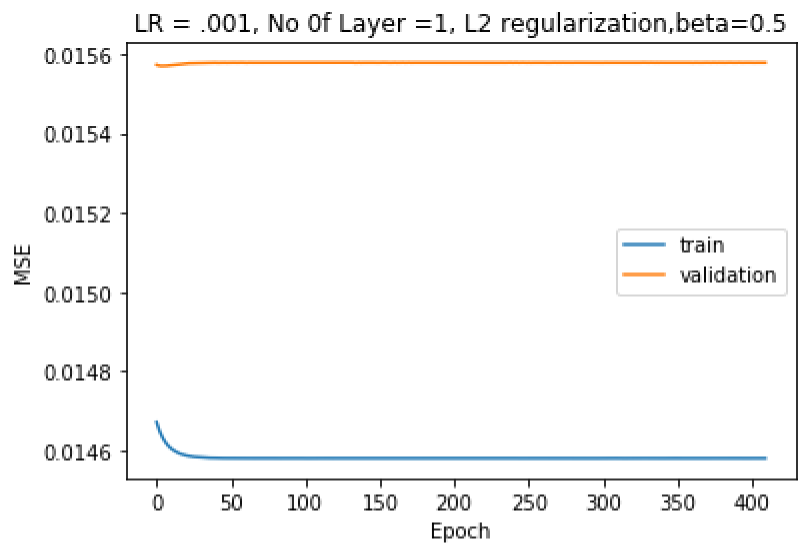
1. At value less that .1:
   Value = .01
   Epoch needed = 300
   Training RMSE: .1120819
   Validation RMSE: .11927
   (epoch showed from 100)

```
plt.title('LR = .001, No Of Layer =1, L2 regularization,beta=0.01')
plt.plot(loss_tr[100:], label = 'train')
plt.plot(loss_va[100:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



1. At value greater than .1:
   Value = .5
   Epoch needed = 400
   Training RMSE: .120748
   Validation RMSE: .12481
   (epoch showed from 190)

```
plt.title('LR = .001, No 0f Layer =1, L2 regularization,beta=0.5')
plt.plot(loss_tr[190:], label = 'train')
plt.plot(loss_va[190:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



4.2.1.3.No of batch tuning:

For one layer NN model No of batch over and below 80 showed worse results than 80.

4.2.1.4.Optimizer tuning:

Among the optimizers provided by tensorflow,Gradient Descent,Adam Optimizer,RMSprop were used here.

- Gradient Descent was really slow
- Adam optimzer was slightly better than Gradient Descent
- RMSprop perfromed best. So RMSprop optimzer was chosen.

4.2.1.5. Activation function selection:

No activation function has been used in this model because applying activation function did not improve the result. Details comments on why use of activation function is not always a good option has been discussed in the discussion section.

4.2.2. Layer 2(1 hidden layer):

4.2.2.1. Summary of 1 hidden layer model:

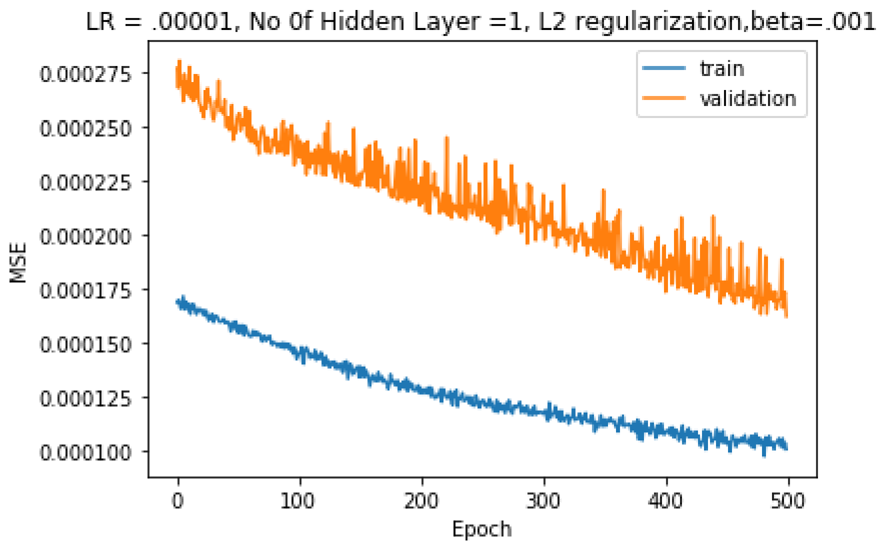
In a one hidden layer based NN:

- The best tuned number of nodes of the hidden layer was 2.
- The 2 layer model was tried with various kinds of parameter tuning such as various range of regularization constant values,learning rate,epoch numbers etc and use of activation function.

4.2.2.2. Mentionable trials on 1 hidden layer model:

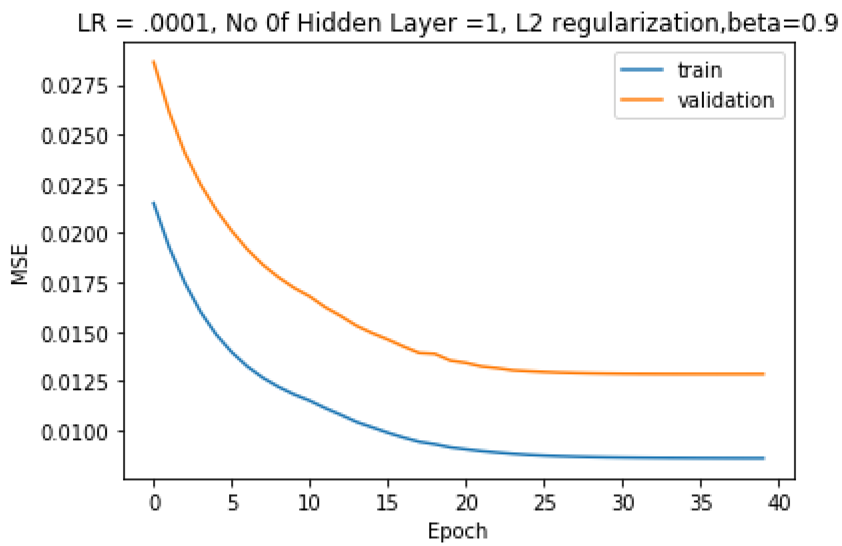
1. Learning rate = 0.00001
   Reg. const = .001
   Epochs = 3000
   Train RMSE: 0.009581
   Validation RMSE: 0.012729
   kaggle score: 0.17266

```
#Plotting
plt.title('LR = .00001, No Of Hidden Layer =1, L2 regularization,beta=.001')
plt.plot(loss_tr[2500:], label = 'train')
plt.plot(loss_va[2500:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



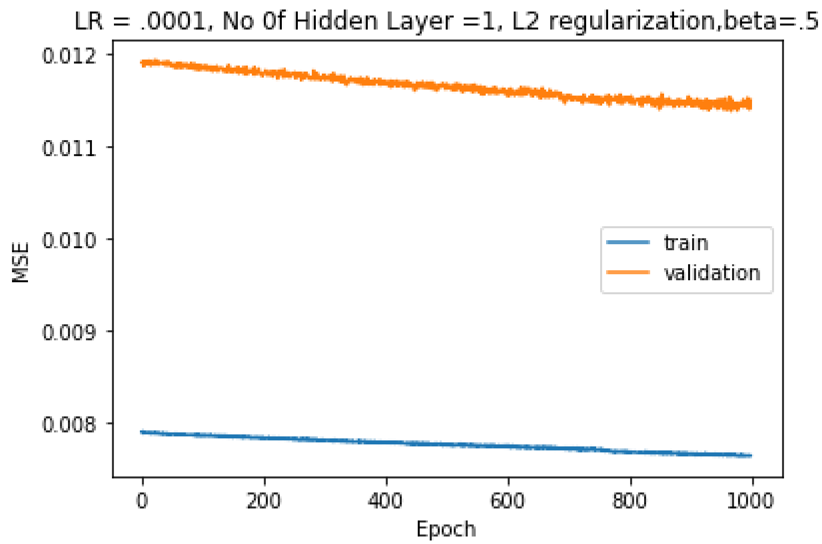
1. Learning rate = 0.0001
   Reg. const = .9
   Epochs = 90
   Train RMSE: 0.0926489
   Validation RMSE: 0.11333814
   kaggle score: .12927

```
#Plotting
plt.title('LR = .0001, No 0f Hidden Layer =1, L2 regularization,beta=0.9')
plt.plot(loss_tr[50:90], label = 'train')
plt.plot(loss_va[50:90],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



1. Learning rate = 0.0001
   Reg. const = .5
   Epochs = 2000
   Train RMSE: 0.08739
   Validation RMSE: 0.107232
   kaggle score: .12904

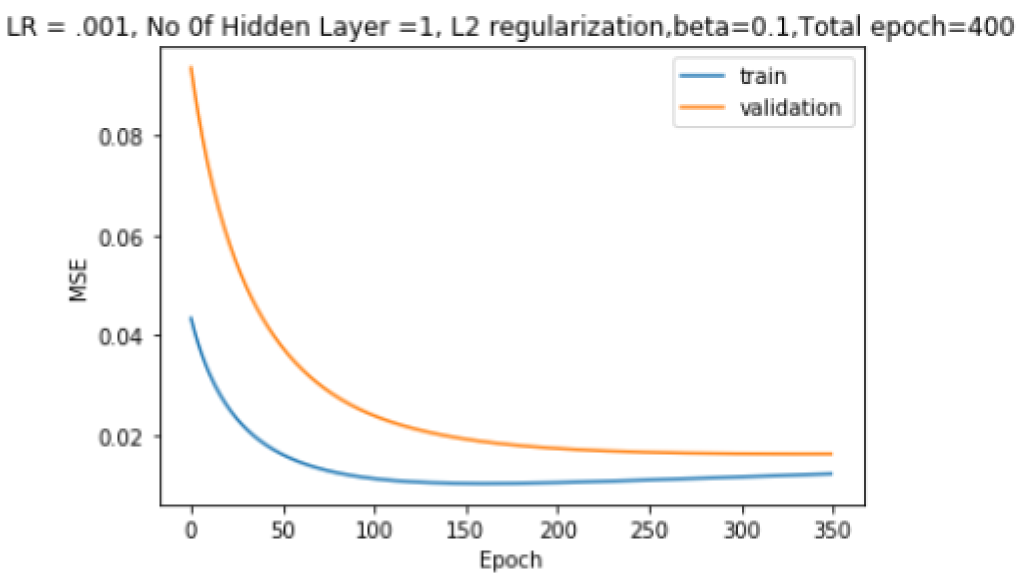
```
#Plotting
plt.title('LR = .0001, No 0f Hidden Layer =1, L2 regularization,beta=.5')
plt.plot(loss_tr[1000:], label = 'train')
plt.plot(loss_va[1000:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



Best result from 1 hidden layer model:

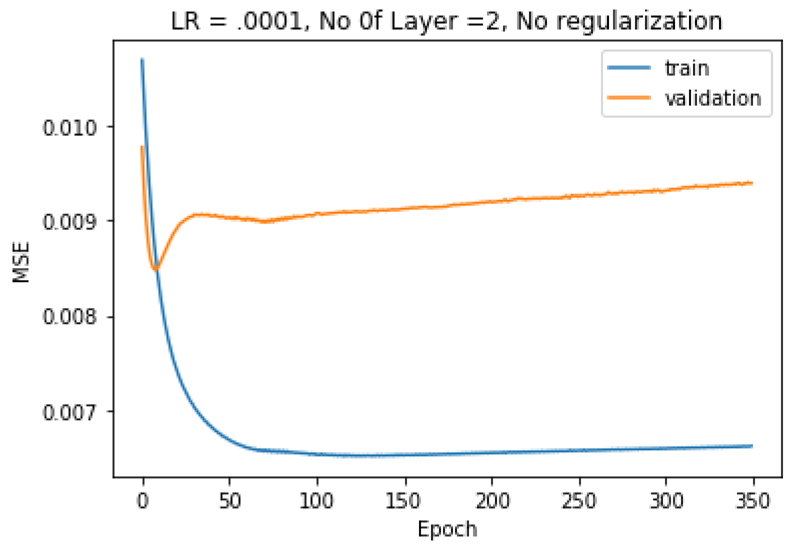
- Use of activation functions did not affect the result significantly. Without relu activation function, the best kaggle score found was 0.12605 at 400 epochs with regularization constant as 0.1 and a learning rate of .0001.
- Epochs more than this were overfitting the model.

```
#Plotting
plt.title('LR = .001, No Of Hidden Layer =1, L2 regularization,beta=0.1,Total epoch=400')
plt.plot(loss_tr[50:], label = 'train')
plt.plot(loss_va[50:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

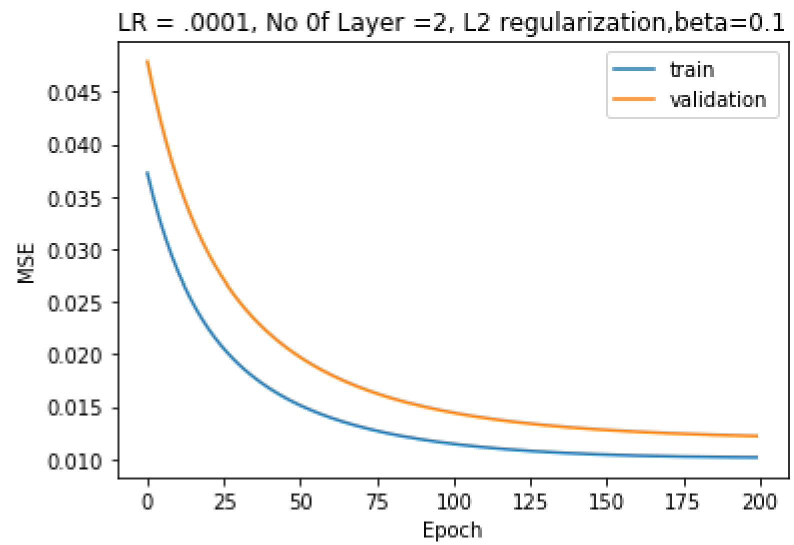


4.2.2.2. Regularization effect:

1. Without any regularization best learning rate was found to be .0001 at around 18 epoch.After that validation loss increased with decreasing training loss. So the model was overfitted.



1. With regularization, after a certain epoch at a certain regularization constant, the training and validation losses remained almost constant. So regularization ensure the model was not overfitting.
   Training RMSE = .10073
   Validationting RMSE = .11041



4.2.3. Best Results from ANN models with higher number of hidden layers

ANN models with higher number of layers has been built and tuning has been done similarly as described above for previous models. Only best result from mentionable models (after proper tuning) are summarized here:

4.2.3.1 ANN model with 2 hidden layers

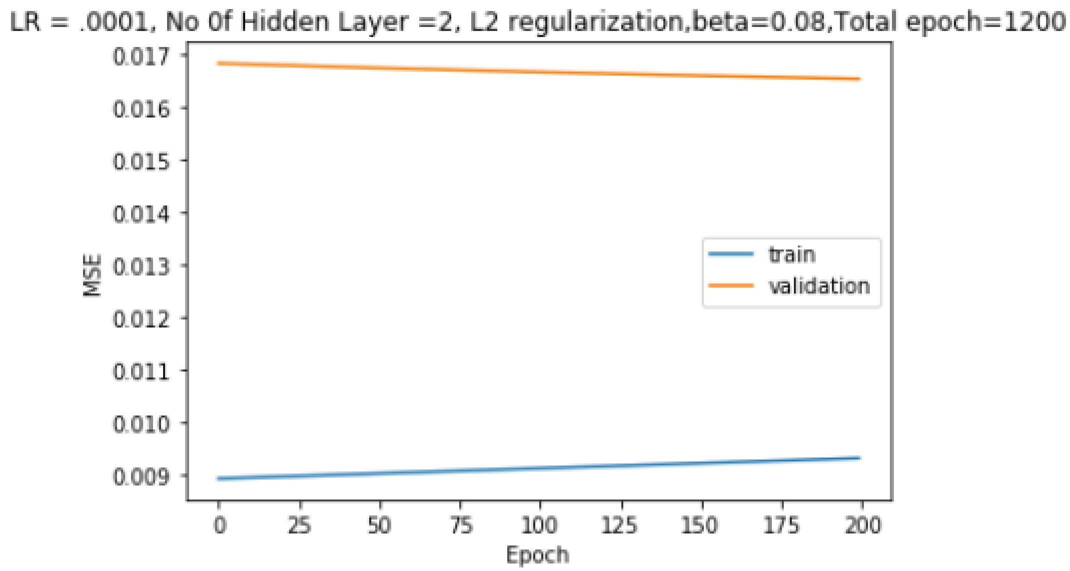
Tuned parameters:

- Number of nodes: Hidden layer 1: Half of Input Layer, Hidden layer 2: 2
- Learning rate: 0.0001
- Regularization constant: 0.08
- Batch size = 60
- Epoch = 1200
- Training RMSE = 0.10220
- Validation RMSE = 0.12443
- Kaggle score = 0.12140

```
print(np.sqrt(loss_tr[-1]),np.sqrt(loss_va[-1]))
```

```
0.09646908331860587 0.12862675239506818
```

```
#Plotting
plt.title('LR = .0001, No Of Hidden Layer =2, L2 regularization,beta=0.08,Total epoch=1200')
plt.plot(loss_tr[1000:], label = 'train')
plt.plot(loss_va[1000:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

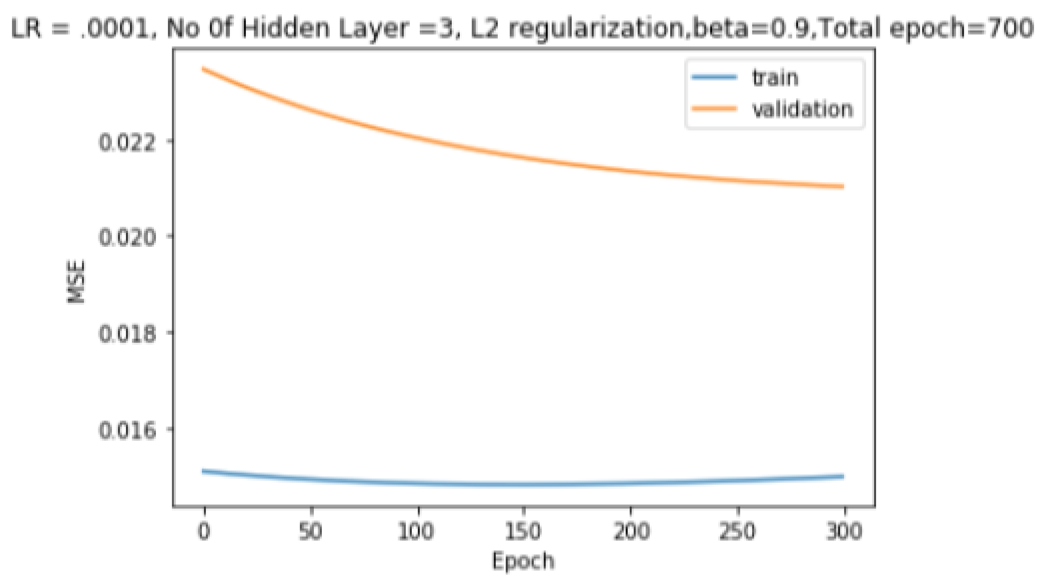


4.2.3.1 2 ANN model with 3 hidden layers

Tuned parameters:

- Number of nodes: Half of previous layer
- Learning rate: 0.0001
- Regularization constant: 0.9
- No of Batch = 80
- Epoch = 550
- Training RMSE = 0.12176
- Validation RMSE = 0.14704
- Kaggle score = 0.14544

```
plt.plot(loss_tr[400:700], label = 'train')
plt.plot(loss_va[400:700],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```

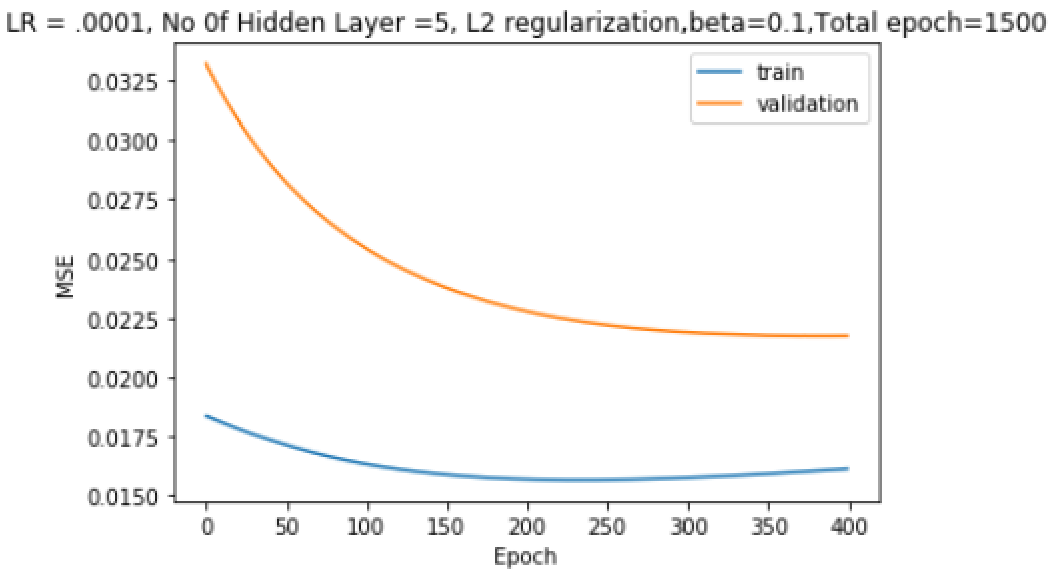


4.2.3.1 2 ANN model with 5 hidden layers

Tuned parameters:

- Number of nodes: Half of previous layer
- Learning rate: 0.0001
- Regularization constant: 0.1
- No of Batch = 80
- Epoch = 450
- Kaggle score = 0.14871
- Training RMSE = 0.12519
- Validation RMSE = 0.14900

```
#Plotting
plt.title('LR = .0001, No Of Hidden Layer =5, L2 regularization,beta=0.1,Total epoch=1500')
plt.plot(loss_tr[200:600], label = 'train')
plt.plot(loss_va[200:600],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



4.3. Final model with tuned hyperparameters

In [38]:
```
LearningRate = .0001
epoch = 1200
reg_const = .08
```

In [39]:
```
no_of_batch=60
no_of_train_batch=int(0.70*len(train_df))

train_data=np.array_split(train_df[:no_of_train_batch],no_of_batch)
train_lb=np.array_split(labels_train[:no_of_train_batch],no_of_batch)
val_data=np.array_split(train_df[no_of_train_batch:],no_of_batch)
val_lb=np.array_split(labels_train[no_of_train_batch:],no_of_batch)

input_feat,sess,out,loss_tr,loss_va=train_loop(LearningRate,epoch,reg_const,train_data,train_lb,val_data,val_
lb)
```

```
epoch: 0 train_loss 115.79189173380534 valid_loss 91.18345578511556
epoch: 1 train_loss 65.8834462483724 valid_loss 46.11759376525879
epoch: 2 train_loss 29.18095194498698 valid_loss 17.48763066927592
epoch: 3 train_loss 10.832053605715434 valid_loss 6.798764157295227
epoch: 4 train_loss 5.631498018900554 valid_loss 4.5139176607131954
epoch: 5 train_loss 3.9787054558595023 valid_loss 3.596071495115757
epoch: 6 train_loss 3.0190464397271475 valid_loss 3.0114154065648715
epoch: 7 train_loss 2.3931628614664078 valid_loss 2.5835654782752195
epoch: 8 train_loss 1.954635156194369 valid_loss 2.238831127186616
epoch: 9 train_loss 1.6259155531724294 valid_loss 1.9495301855107148
epoch: 10 train_loss 1.3681620692213377 valid_loss 1.7037312837938468
epoch: 11 train_loss 1.160637237628301 valid_loss 1.4940541038910549
epoch: 12 train_loss 0.9909056524435679 valid_loss 1.3147395508984725
epoch: 13 train_loss 0.850621385872364 valid_loss 1.160862253109614
epoch: 14 train_loss 0.7337289785345396 valid_loss 1.0282810849448045
epoch: 15 train_loss 0.6356332542995612 valid_loss 0.9134979143738746
epoch: 16 train_loss 0.5527847732106844 valid_loss 0.8137888312339783
epoch: 17 train_loss 0.48242192417383195 valid_loss 0.726892585804065
epoch: 18 train_loss 0.4223865022261937 valid_loss 0.650902779897054
epoch: 19 train_loss 0.3709841199219227 valid_loss 0.5843253083527088
epoch: 20 train_loss 0.32686804234981537 valid_loss 0.5259307506183784
epoch: 21 train_loss 0.28894557344416777 valid_loss 0.47469817399978637
epoch: 22 train_loss 0.2563123396287362 valid_loss 0.4297581076622009
epoch: 23 train_loss 0.22820510069529215 valid_loss 0.390324659148852
epoch: 24 train_loss 0.20397035479545594 valid_loss 0.35571363940835
epoch: 25 train_loss 0.18305023685097693 valid_loss 0.3253262985497713
epoch: 26 train_loss 0.1649635742108027 valid_loss 0.29862733781337736
epoch: 27 train_loss 0.1492949012046059 valid_loss 0.2751526560013493
epoch: 28 train_loss 0.13568770543982586 valid_loss 0.2544935200984279
epoch: 29 train_loss 0.12384170641501745 valid_loss 0.2362876276796063
epoch: 30 train_loss 0.11349916327744722 valid_loss 0.22021398848543564
epoch: 31 train_loss 0.10444173316160837 valid_loss 0.20598921943455933
epoch: 32 train_loss 0.09648396534224352 valid_loss 0.19336392718056838
epoch: 33 train_loss 0.08946819361299277 valid_loss 0.18211995648841064
epoch: 34 train_loss 0.08326043887063861 valid_loss 0.17206763674815495
epoch: 35 train_loss 0.0777467917650938 valid_loss 0.16304256149257224
epoch: 36 train_loss 0.07282912051305175 valid_loss 0.15490395526091258
epoch: 37 train_loss 0.06842270269989967 valid_loss 0.1475325779678921
epoch: 38 train_loss 0.0644659338829418 valid_loss 0.14082662764315804
epoch: 39 train_loss 0.06089718192815781 valid_loss 0.13469926125059525
epoch: 40 train_loss 0.05766576298822959 valid_loss 0.12907681356494624
epoch: 41 train_loss 0.05472891268630822 valid_loss 0.12389690279960633
epoch: 42 train_loss 0.05205035579080383 valid_loss 0.11910669980570673
```

```
epoch: 43 train_loss 0.049599209365745384 valid_loss 0.11466154692073663
epoch: 44 train_loss 0.047349201825757824 valid_loss 0.11052353326231242
epoch: 45 train_loss 0.045277967133248845 valid_loss 0.10666030781964461
epoch: 46 train_loss 0.043365314571807784 valid_loss 0.10304419297414522
epoch: 47 train_loss 0.04159488131602605 valid_loss 0.09965131726736824
epoch: 48 train_loss 0.03995222096952299 valid_loss 0.09646087923708062
epoch: 49 train_loss 0.03842471648628513 valid_loss 0.09345474846971531
epoch: 50 train_loss 0.03700146018527448 valid_loss 0.09061707756482065
epoch: 51 train_loss 0.03567286605636279 valid_loss 0.08793436468889317
epoch: 52 train_loss 0.034430549511065084 valid_loss 0.08539489392812054
epoch: 53 train_loss 0.03326703036824862 valid_loss 0.08298824190472563
epoch: 54 train_loss 0.032175706777100764 valid_loss 0.08070499234211942
epoch: 55 train_loss 0.031150720408186318 valid_loss 0.07853663655308386
epoch: 56 train_loss 0.030186954078574975 valid_loss 0.07647545547224581
epoch: 57 train_loss 0.02928009582683444 valid_loss 0.07451443849131464
epoch: 58 train_loss 0.028425629436969756 valid_loss 0.07264718324877321
epoch: 59 train_loss 0.027618576136107246 valid_loss 0.0708678194321692
epoch: 60 train_loss 0.026855859672650695 valid_loss 0.06917102073008816
epoch: 61 train_loss 0.026134554607172806 valid_loss 0.06755182671671113
epoch: 62 train_loss 0.02545187814782063 valid_loss 0.06600565185459951
epoch: 63 train_loss 0.02480521691031754 valid_loss 0.06452825913826625
epoch: 64 train_loss 0.024192198583235344 valid_loss 0.06311571344267577
epoch: 65 train_loss 0.023610596521757544 valid_loss 0.061764349299483004
epoch: 66 train_loss 0.023058372673889 valid_loss 0.06047073023704191
epoch: 67 train_loss 0.022533666932334502 valid_loss 0.05923168240891149
epoch: 68 train_loss 0.022034744406118988 valid_loss 0.05804417975402127
epoch: 69 train_loss 0.021559991392617425 valid_loss 0.05690550407549987
epoch: 70 train_loss 0.021107934407579403 valid_loss 0.0558129837969318
epoch: 71 train_loss 0.020677176335205636 valid_loss 0.05476420785610874
epoch: 72 train_loss 0.020266408434448144 valid_loss 0.053756880876608196
epoch: 73 train_loss 0.019874328087704878 valid_loss 0.052788866427727045
epoch: 74 train_loss 0.01949928174726665 valid_loss 0.05185814472691466
epoch: 75 train_loss 0.019139646226540206 valid_loss 0.05096286797585587
epoch: 76 train_loss 0.01879781186580658 valid_loss 0.05010125685172776
epoch: 77 train_loss 0.01847139235275487 valid_loss 0.0492716661732023
epoch: 78 train_loss 0.01815884980993966 valid_loss 0.04847251232713460
epoch: 79 train_loss 0.017859598629487057 valid_loss 0.047702366043813525
epoch: 80 train_loss 0.017572938472342987 valid_loss 0.046959826109620434
epoch: 81 train_loss 0.01729819003958255 valid_loss 0.04624361243719856
epoch: 82 train_loss 0.017034702748060227 valid_loss 0.04555251324394097
epoch: 83 train_loss 0.01678190358603994 valid_loss 0.04488538215712955
epoch: 84 train_loss 0.016539219060602288 valid_loss 0.044241151842288676
epoch: 85 train_loss 0.01630613352948179 valid_loss 0.04361874976505835
```

```
epoch: 86 train_loss 0.016082149053302904 valid_loss 0.043017283788261314
epoch: 87 train_loss 0.015866798845430217 valid_loss 0.04243578706712772
epoch: 88 train_loss 0.015659642118650178 valid_loss 0.0418734315627565
epoch: 89 train_loss 0.015460261376574636 valid_loss 0.04132942048211893
epoch: 90 train_loss 0.015268230554647743 valid_loss 0.04080299731964866
epoch: 91 train_loss 0.015083158489627142 valid_loss 0.04029345114249736
epoch: 92 train_loss 0.014904786514428755 valid_loss 0.039800121223864456
epoch: 93 train_loss 0.014733036879139643 valid_loss 0.039322451253732044
epoch: 94 train_loss 0.014567713377376397 valid_loss 0.038859791176704066
epoch: 95 train_loss 0.014408627098115781 valid_loss 0.038411702044929066
epoch: 96 train_loss 0.014255817506151895 valid_loss 0.03797765041623886
epoch: 97 train_loss 0.014108560793101788 valid_loss 0.037557153493010746
epoch: 98 train_loss 0.01396623244509101 valid_loss 0.03714976526874428
epoch: 99 train_loss 0.01382873572098712 valid_loss 0.036755032445459315
epoch: 100 train_loss 0.013695876494360467 valid_loss 0.036372497689444575
epoch: 101 train_loss 0.013567449206796785 valid_loss 0.0360017928488863
epoch: 102 train_loss 0.013443216759090623 valid_loss 0.0356424944436488
epoch: 103 train_loss 0.01332295297179371 valid_loss 0.035294185241218655
epoch: 104 train_loss 0.0132064172920460 valid_loss 0.03495609392800058
epoch: 105 train_loss 0.013093302818015217 valid_loss 0.03462432112234334
epoch: 106 train_loss 0.012982790812384338 valid_loss 0.034248686011414974
epoch: 107 train_loss 0.012877062242478133 valid_loss 0.03345544141096373
epoch: 108 train_loss 0.012764768255874515 valid_loss 0.03349592396989465
epoch: 109 train_loss 0.012671521018880109 valid_loss 0.03283568137946228
epoch: 110 train_loss 0.012568226914542418 valid_loss 0.032890027451018496
epoch: 111 train_loss 0.012482020371438314 valid_loss 0.0322295087040402
epoch: 112 train_loss 0.012383117286177973 valid_loss 0.03229786461452022
epoch: 113 train_loss 0.012303383217658847 valid_loss 0.0316601494133162
epoch: 114 train_loss 0.012208224487646172 valid_loss 0.03171396961746117
epoch: 115 train_loss 0.012133508218297114 valid_loss 0.031132490440116574
epoch: 116 train_loss 0.012043220080280055 valid_loss 0.031145111246344944
epoch: 117 train_loss 0.011972724479467919 valid_loss 0.030632735098091265
epoch: 118 train_loss 0.011887842354675134 valid_loss 0.030610718744962167
epoch: 119 train_loss 0.011821168843501558 valid_loss 0.03015562651368479
epoch: 120 train_loss 0.011741390711783121 valid_loss 0.0301094753590102
epoch: 121 train_loss 0.011678195558488369 valid_loss 0.02970054477530842
epoch: 122 train_loss 0.011603216722141952 valid_loss 0.02963501171519359
epoch: 123 train_loss 0.011543056089431047 valid_loss 0.02926696326661234
epoch: 124 train_loss 0.011472766772688677 valid_loss 0.02918432791872571
epoch: 125 train_loss 0.011415187665261329 valid_loss 0.028853842053407183
epoch: 126 train_loss 0.011349404230713844 valid_loss 0.02875560096775492
epoch: 127 train_loss 0.011294212095284214 valid_loss 0.028459876325602332
epoch: 128 train_loss 0.011232977651525288 valid_loss 0.028347582280791053
```

```
epoch: 129 train_loss 0.011180502024944871 valid_loss 0.028083724904960642
epoch: 130 train_loss 0.011122977562869589 valid_loss 0.027959242715345074
epoch: 131 train_loss 0.011071930907201022 valid_loss 0.027724017012709132
epoch: 132 train_loss 0.011017723470771065 valid_loss 0.02758973342133686
epoch: 133 train_loss 0.010968569646744678 valid_loss 0.027379365370143204
epoch: 134 train_loss 0.010917594105315705 valid_loss 0.027238197108575453
epoch: 135 train_loss 0.010870360819778095 valid_loss 0.027048510615713894
epoch: 136 train_loss 0.010822326639511933 valid_loss 0.026903626532293855
epoch: 137 train_loss 0.010777014641401668 valid_loss 0.02673050680120165
epoch: 138 train_loss 0.010731629542230317 valid_loss 0.026584887826660027
epoch: 139 train_loss 0.01068823564176758 valid_loss 0.026424684829544277
epoch: 140 train_loss 0.010645224231605728 valid_loss 0.026280762200864654
epoch: 141 train_loss 0.010603727943574389 valid_loss 0.026130597359345604
epoch: 142 train_loss 0.010562866216059775 valid_loss 0.025990028687131902
epoch: 143 train_loss 0.010523210694858183 valid_loss 0.02584788095749294
epoch: 144 train_loss 0.010484322583458076 valid_loss 0.025711676540474095
epoch: 145 train_loss 0.010446444832875082 valid_loss 0.025576162431389094
epoch: 146 train_loss 0.010409373964648694 valid_loss 0.025444723836456736
epoch: 147 train_loss 0.010373201019441088 valid_loss 0.025315022782888264
epoch: 148 train_loss 0.01033782676095143 valid_loss 0.02518844618462026
epoch: 149 train_loss 0.010303264627388368 valid_loss 0.02506402493454516
epoch: 150 train_loss 0.01026947171970581 valid_loss 0.02494223304092884
epoch: 151 train_loss 0.010236445011105388 valid_loss 0.02482266805600375
epoch: 152 train_loss 0.01020414229715243 valid_loss 0.024705481769827505
epoch: 153 train_loss 0.010172559254957985 valid_loss 0.024590490160820384
epoch: 154 train_loss 0.010141656807779025 valid_loss 0.02447770731523633
epoch: 155 train_loss 0.010111431952100246 valid_loss 0.02436705098565047
epoch: 156 train_loss 0.010081858350895345 valid_loss 0.024258487958771488
epoch: 157 train_loss 0.010052916038936625 valid_loss 0.0241519353313682
epoch: 158 train_loss 0.010024597402662039 valid_loss 0.02404738968471065
epoch: 159 train_loss 0.009996868049105009 valid_loss 0.02394476344731326
epoch: 160 train_loss 0.009969727706629784 valid_loss 0.02384403465160479
epoch: 161 train_loss 0.009943151652502516 valid_loss 0.023745139913323023
epoch: 162 train_loss 0.009917128287876645 valid_loss 0.023648056869084635
epoch: 163 train_loss 0.00989163904838885 valid_loss 0.02355272090062499
epoch: 164 train_loss 0.009866673860233276 valid_loss 0.0234591054613702
epoch: 165 train_loss 0.009842214562619725 valid_loss 0.02336719259619713
epoch: 166 train_loss 0.009818251768592745 valid_loss 0.023276905839641888
epoch: 167 train_loss 0.009794764736822496 valid_loss 0.02318821904870371
epoch: 168 train_loss 0.009771752229426056 valid_loss 0.02310110246374582
epoch: 169 train_loss 0.009749192146894832 valid_loss 0.023015513174080602
epoch: 170 train_loss 0.009727091252959024 valid_loss 0.022931422875262796
epoch: 171 train_loss 0.009705416140301773 valid_loss 0.02284880867615963
```

```
epoch: 172 train_loss 0.009684170813610156 valid_loss 0.022767596773337572
epoch: 173 train_loss 0.009663335629738867 valid_loss 0.022687788773328065
epoch: 174 train_loss 0.00964290783352529 valid_loss 0.022609356817944595
epoch: 175 train_loss 0.009622873559904595 valid_loss 0.02253225169843063
epoch: 176 train_loss 0.009603227605111897 valid_loss 0.022456450717678916
epoch: 177 train_loss 0.00958395650377497 valid_loss 0.02238191366971781
epoch: 178 train_loss 0.009565060655586422 valid_loss 0.022308642673306168
epoch: 179 train_loss 0.00954651488379265 valid_loss 0.02223656487185508
epoch: 180 train_loss 0.009528329704577725 valid_loss 0.022165687463711947
epoch: 181 train_loss 0.009510483604390173 valid_loss 0.022095970118728776
epoch: 182 train_loss 0.00949297782111292 valid_loss 0.022027385355128597
epoch: 183 train_loss 0.00947580262630557 valid_loss 0.02195992460474372
epoch: 184 train_loss 0.009458951675333083 valid_loss 0.021893551293760537
epoch: 185 train_loss 0.00944240902317688 valid_loss 0.021828233931834497
epoch: 186 train_loss 0.009426182438619434 valid_loss 0.021763962953506657
epoch: 187 train_loss 0.009410254859055082 valid_loss 0.02170070050827538
epoch: 188 train_loss 0.009394626152546456 valid_loss 0.021638430643361063
epoch: 189 train_loss 0.009379287933309872 valid_loss 0.02157714501178513
epoch: 190 train_loss 0.009364228505486 valid_loss 0.021516798157244922
epoch: 191 train_loss 0.009349453961476684 valid_loss 0.021457390580326318
epoch: 192 train_loss 0.009334955527447165 valid_loss 0.02139889405031378
epoch: 193 train_loss 0.00932072135231768 valid_loss 0.021341299638152122
epoch: 194 train_loss 0.009306751552503555 valid_loss 0.021284565054035434
epoch: 195 train_loss 0.009293040994089096 valid_loss 0.0212286810313041
epoch: 196 train_loss 0.009279587251755098 valid_loss 0.021173628765003134
epoch: 197 train_loss 0.0092663817301703 valid_loss 0.021119410024645426
epoch: 198 train_loss 0.009253429862049719 valid_loss 0.0210659913214234
epoch: 199 train_loss 0.009240717565019926 valid_loss 0.021013345022220166
epoch: 200 train_loss 0.009228254986616473 valid_loss 0.020961467238763968
epoch: 201 train_loss 0.009216029837261885 valid_loss 0.020910354043977956
epoch: 202 train_loss 0.009204059171800812 valid_loss 0.020859966240823268
epoch: 203 train_loss 0.00919233742946138 valid_loss 0.020810304391973962
epoch: 204 train_loss 0.00918086275535946 valid_loss 0.020761337499910344
epoch: 205 train_loss 0.009169634835173687 valid_loss 0.02071307294924433
epoch: 206 train_loss 0.009158650320023299 valid_loss 0.0206655093585141
epoch: 207 train_loss 0.009147885934604952 valid_loss 0.02061857448813195
epoch: 208 train_loss 0.009137331558546673 valid_loss 0.020572302769869565
epoch: 209 train_loss 0.009126935480162501 valid_loss 0.02052667043171823
epoch: 210 train_loss 0.009116704051848501 valid_loss 0.02048166337578247
epoch: 211 train_loss 0.009106617514044046 valid_loss 0.02043726834623764
epoch: 212 train_loss 0.009096672886516898 valid_loss 0.020393473383349677
epoch: 213 train_loss 0.00908689018106088 valid_loss 0.020350273100969693
epoch: 214 train_loss 0.009077249413045744 valid_loss 0.020307650335598736
```

```
epoch: 215 train_loss 0.009067770462327948 valid_loss 0.020265584951266645
epoch: 216 train_loss 0.009058444376569242 valid_loss 0.020224071709283937
epoch: 217 train_loss 0.009049278862463931 valid_loss 0.020183107846726975
epoch: 218 train_loss 0.009040268642517427 valid_loss 0.020142692688386886
epoch: 219 train_loss 0.009031416370999069 valid_loss 0.020102787646465003
epoch: 220 train_loss 0.009022719560501475 valid_loss 0.020063391611135253
epoch: 221 train_loss 0.009014172852039338 valid_loss 0.020024506266539295
epoch: 222 train_loss 0.009005780308507383 valid_loss 0.01998613424754391
epoch: 223 train_loss 0.008997536621366937 valid_loss 0.019948210866035272
epoch: 224 train_loss 0.008989435228674363 valid_loss 0.019910791205863157
epoch: 225 train_loss 0.008981485303957015 valid_loss 0.019873842701781542
epoch: 226 train_loss 0.008973675993426392 valid_loss 0.019837336890244237
epoch: 227 train_loss 0.008966003455376874 valid_loss 0.019801289095388103
epoch: 228 train_loss 0.00895847543142736 valid_loss 0.019765684268592546
epoch: 229 train_loss 0.008951078995596617 valid_loss 0.019730526596928637
epoch: 230 train_loss 0.008943821889503548 valid_loss 0.019695785692116868
epoch: 231 train_loss 0.008936691392833988 valid_loss 0.01966144881832103
epoch: 232 train_loss 0.008929693920072168 valid_loss 0.019627545812788107
epoch: 233 train_loss 0.008922822043920557 valid_loss 0.019594032038003206
epoch: 234 train_loss 0.00891607654436181 valid_loss 0.019560930063016714
epoch: 235 train_loss 0.008909449334411571 valid_loss 0.01952821626327932
epoch: 236 train_loss 0.008902954589575528 valid_loss 0.019495864406538506
epoch: 237 train_loss 0.00889657021034509 valid_loss 0.019463903592744223
epoch: 238 train_loss 0.008890308129290739 valid_loss 0.01943232478806749
epoch: 239 train_loss 0.008884157271434863 valid_loss 0.019401107639229545
epoch: 240 train_loss 0.008878126084649314 valid_loss 0.019370232285776487
epoch: 241 train_loss 0.008872206590604037 valid_loss 0.01933972960105166
epoch: 242 train_loss 0.008866392421380927 valid_loss 0.019309567733823012
epoch: 243 train_loss 0.008860688998053472 valid_loss 0.019279729031647246
epoch: 244 train_loss 0.00885509188907842 valid_loss 0.01925023183769857
epoch: 245 train_loss 0.008849600073881447 valid_loss 0.019221092492807658
epoch: 246 train_loss 0.008844212275774529 valid_loss 0.019192251209945728
epoch: 247 train_loss 0.008838923752773554 valid_loss 0.01916371986347561
epoch: 248 train_loss 0.008833742712158709 valid_loss 0.019135528050052624
epoch: 249 train_loss 0.00882865188177675 valid_loss 0.019107624252016345
epoch: 250 train_loss 0.008823658572509885 valid_loss 0.01908003477146849
epoch: 251 train_loss 0.008818763180170209 valid_loss 0.019052742406105
epoch: 252 train_loss 0.008813964866567403 valid_loss 0.019025745813269167
epoch: 253 train_loss 0.008809254826822628 valid_loss 0.01899903469796603
epoch: 254 train_loss 0.008804638342310985 valid_loss 0.018972623080480845
epoch: 255 train_loss 0.008800110321802397 valid_loss 0.018946467025671154
epoch: 256 train_loss 0.008795668235203872 valid_loss 0.018920598400291054
epoch: 257 train_loss 0.008791316878826668 valid_loss 0.018895006474728384
```

```
epoch: 258 train_loss 0.008787052915431559 valid_loss 0.0188696938372838
epoch: 259 train_loss 0.008782867951473842 valid_loss 0.018844623766684283
epoch: 260 train_loss 0.008778772801936915 valid_loss 0.018819813368221125
epoch: 261 train_loss 0.008774756896309554 valid_loss 0.018795264582149685
epoch: 262 train_loss 0.008770821724707882 valid_loss 0.01877098171195636
epoch: 263 train_loss 0.008766965905670077 valid_loss 0.018746927027435353
epoch: 264 train_loss 0.008763187972363085 valid_loss 0.01872312429283435
epoch: 265 train_loss 0.008759486748992155 valid_loss 0.018699575557063024
epoch: 266 train_loss 0.008755866205319762 valid_loss 0.018676245006887863
epoch: 267 train_loss 0.008752318937331438 valid_loss 0.018653162758952627
epoch: 268 train_loss 0.008748842771941176 valid_loss 0.018630307268661758
epoch: 269 train_loss 0.008745447720866651 valid_loss 0.01860766438379263
epoch: 270 train_loss 0.008742116232557844 valid_loss 0.01858526236998538
epoch: 271 train_loss 0.008738857325321684 valid_loss 0.01856308737381672
epoch: 272 train_loss 0.008735670700358848 valid_loss 0.01854111294572552
epoch: 273 train_loss 0.008732549950946123 valid_loss 0.018519355457586546
epoch: 274 train_loss 0.008729503757786005 valid_loss 0.018497837144726265
epoch: 275 train_loss 0.008726518407153586 valid_loss 0.018476503284182398
epoch: 276 train_loss 0.008723600406665356 valid_loss 0.018455383973196147
epoch: 277 train_loss 0.0087207474008513 valid_loss 0.018434454197995366
epoch: 278 train_loss 0.008717962358302127 valid_loss 0.018413741184243312
epoch: 279 train_loss 0.008715236664284021 valid_loss 0.018393217058231433
epoch: 280 train_loss 0.008712575087944667 valid_loss 0.018372893422686807
epoch: 281 train_loss 0.008709976449608803 valid_loss 0.018352768449888875
epoch: 282 train_loss 0.00870743573953708 valid_loss 0.018332823152498654
epoch: 283 train_loss 0.00870495723405232 valid_loss 0.018313070704850058
epoch: 284 train_loss 0.008702535310294478 valid_loss 0.018293514750742663
epoch: 285 train_loss 0.008700172234481822 valid_loss 0.018274131483243156
epoch: 286 train_loss 0.008697869225094716 valid_loss 0.018254918919410557
epoch: 287 train_loss 0.008695626068705073 valid_loss 0.018235893923944483
epoch: 288 train_loss 0.008693433071797093 valid_loss 0.018217047082725913
epoch: 289 train_loss 0.008691297211529066 valid_loss 0.01819836413099741
epoch: 290 train_loss 0.008689217207332452 valid_loss 0.018179868219885976
epoch: 291 train_loss 0.008687189353319505 valid_loss 0.01816153551529472
epoch: 292 train_loss 0.008685216761659832 valid_loss 0.01814338118225957
epoch: 293 train_loss 0.008683295284087459 valid_loss 0.018125369485157233
epoch: 294 train_loss 0.008681429367667685 valid_loss 0.01810753591125831
epoch: 295 train_loss 0.008679605306436617 valid_loss 0.018089870759285986
epoch: 296 train_loss 0.00867783868064483 valid_loss 0.01807235061035802
epoch: 297 train_loss 0.008676118857692927 valid_loss 0.018054996027300756
epoch: 298 train_loss 0.008674453594721854 valid_loss 0.018037779776689907
epoch: 299 train_loss 0.008672829845454543 valid_loss 0.01802073240978643
epoch: 300 train_loss 0.008671257361614455 valid_loss 0.01800383795052767
```

```
epoch: 301 train_loss 0.00866973433488359 valid_loss 0.017987086068994056
epoch: 302 train_loss 0.008668252309629072 valid_loss 0.017970492240662377
epoch: 303 train_loss 0.00866681761884441 valid_loss 0.017954033485148103
epoch: 304 train_loss 0.008665435602112363 valid_loss 0.017937714389214914
epoch: 305 train_loss 0.00866409121469284 valid_loss 0.01792156231046343 8
epoch: 306 train_loss 0.00866279297042638 valid_loss 0.017905535948618004
epoch: 307 train_loss 0.008661536550304542 valid_loss 0.017889645175697903
epoch: 308 train_loss 0.008660326929142077 valid_loss 0.017873890271099906
epoch: 309 train_loss 0.008659161732066423 valid_loss 0.01785828711775442
epoch: 310 train_loss 0.008658033795654774 valid_loss 0.01784280432232966
epoch: 311 train_loss 0.00865695036482066 valid_loss 0.017827479611150922
epoch: 312 train_loss 0.008655905568351349 valid_loss 0.01781226065553104
epoch: 313 train_loss 0.008654905188207825 valid_loss 0.017797183338552715
epoch: 314 train_loss 0.008653942006640136 valid_loss 0.017782218071321645
epoch: 315 train_loss 0.008653018076438456 valid_loss 0.01776739050401375
epoch: 316 train_loss 0.008652137933919828 valid_loss 0.017752700252458453
epoch: 317 train_loss 0.008651297019484142 valid_loss 0.017738111282233147
epoch: 318 train_loss 0.008650487242266535 valid_loss 0.017723672254942358
epoch: 319 train_loss 0.008649722463451326 valid_loss 0.01770933356989796
epoch: 320 train_loss 0.008648990389580528 valid_loss 0.017695132658506432
epoch: 321 train_loss 0.00864829250688975 valid_loss 0.017681046396804352
epoch: 322 train_loss 0.008647640057218572 valid_loss 0.017667071043979375
epoch: 323 train_loss 0.008647020304730782 valid_loss 0.017653202226695915
epoch: 324 train_loss 0.008646438476474335 valid_loss 0.017639468734463055
epoch: 325 train_loss 0.008645891887135803 valid_loss 0.01762583724145467
epoch: 326 train_loss 0.008645376663965483 valid_loss 0.01761231425528725
epoch: 327 train_loss 0.008644899128315349 valid_loss 0.01759892109548673
epoch: 328 train_loss 0.008644456191298862 valid_loss 0.017585614525402585
epoch: 329 train_loss 0.008644044958055019 valid_loss 0.017572429174712547
epoch: 330 train_loss 0.00864366850970934 valid_loss 0.01755934756171579
epoch: 331 train_loss 0.008643326376720021 valid_loss 0.01754637686535716
epoch: 332 train_loss 0.008643015598257383 valid_loss 0.017533498362172394
epoch: 333 train_loss 0.008642738064130148 valid_loss 0.017520731221884488
epoch: 334 train_loss 0.008642495179083198 valid_loss 0.017508079969168953
epoch: 335 train_loss 0.008642285375390201 valid_loss 0.017495502593616645
epoch: 336 train_loss 0.00864210562237228 valid_loss 0.017483054908613363
epoch: 337 train_loss 0.008641956932842731 valid_loss 0.017470676444160442
epoch: 338 train_loss 0.008641840525281925 valid_loss 0.017458428659786782
epoch: 339 train_loss 0.00864174811479946 valid_loss 0.017446268706892927
epoch: 340 train_loss 0.00864168982564782 valid_loss 0.017434195747288564
epoch: 341 train_loss 0.008641668284932772 valid_loss 0.01742221670768534
epoch: 342 train_loss 0.00864167323646446 valid_loss 0.017410318999706458
epoch: 343 train_loss 0.008641708207627137 valid_loss 0.0173985235276632
```

```
epoch: 344 train_loss 0.00864176956238225 valid_loss 0.01738682861129443
epoch: 345 train_loss 0.008641864475794136 valid_loss 0.017375219756892 57
epoch: 346 train_loss 0.008641991690577317 valid_loss 0.017363704973831773
epoch: 347 train_loss 0.008642150201679518 valid_loss 0.017352267086971553
epoch: 348 train_loss 0.008642347494605929 valid_loss 0.01734092638362199
epoch: 349 train_loss 0.00864257498178631 valid_loss 0.0173296721962591
epoch: 350 train_loss 0.00864284229464829 valid_loss 0.01731848772227143 4
epoch: 351 train_loss 0.008643152297008783 valid_loss 0.017307397913342962
epoch: 352 train_loss 0.00864349255959193 valid_loss 0.01729640350289022
epoch: 353 train_loss 0.00864386756438762 valid_loss 0.01728548629131789
epoch: 354 train_loss 0.008644279003298531 valid_loss 0.017274657799862324
epoch: 355 train_loss 0.008644714454809825 valid_loss 0.017263909964822233
epoch: 356 train_loss 0.008645169076044113 valid_loss 0.017253219487611205
epoch: 357 train_loss 0.008645635369854668 valid_loss 0.017242643403975914
epoch: 358 train_loss 0.00864611145031328 valid_loss 0.01723214238882064 8
epoch: 359 train_loss 0.00864659301781406 valid_loss 0.01722170419525355
epoch: 360 train_loss 0.008647086983546614 valid_loss 0.017211361470011375
epoch: 361 train_loss 0.008647591577998052 valid_loss 0.017201102781109513
epoch: 362 train_loss 0.008648106665350496 valid_loss 0.01719090568755443
epoch: 363 train_loss 0.00864863912962998 valid_loss 0.01718078602571040 3
epoch: 364 train_loss 0.0086491825679938 valid_loss 0.017170745634939522
epoch: 365 train_loss 0.008649740430215994 valid_loss 0.017160788422916084
epoch: 366 train_loss 0.00865032048119853 valid_loss 0.01715089750553792
epoch: 367 train_loss 0.008650913337866465 valid_loss 0.017141082716019204
epoch: 368 train_loss 0.00865152032347396 valid_loss 0.01713133760883162 4
epoch: 369 train_loss 0.008652151752418529 valid_loss 0.017121669421127687
epoch: 370 train_loss 0.008652797112396608 valid_loss 0.01711206304219862
epoch: 371 train_loss 0.008653463524145385 valid_loss 0.017102528468240053
epoch: 372 train_loss 0.008654144635268797 valid_loss 0.01709307680527369
epoch: 373 train_loss 0.008654843977031609 valid_loss 0.017083682748489083
epoch: 374 train_loss 0.008655564778018742 valid_loss 0.01707435291415701
epoch: 375 train_loss 0.008656302245799451 valid_loss 0.01706508526112884
epoch: 376 train_loss 0.008657059023001542 valid_loss 0.017055896778280535
epoch: 377 train_loss 0.008657826386236895 valid_loss 0.017046775222600747
epoch: 378 train_loss 0.008658619481138885 valid_loss 0.01703771153697744
epoch: 379 train_loss 0.008659425160537164 valid_loss 0.017028724138314525
epoch: 380 train_loss 0.008660250075627119 valid_loss 0.017019795358646662
epoch: 381 train_loss 0.00866109988419339 valid_loss 0.01701092178312440 7
epoch: 382 train_loss 0.008661953747893374 valid_loss 0.01700212622138982
epoch: 383 train_loss 0.008662831767772635 valid_loss 0.016993379798562577
epoch: 384 train_loss 0.008663727168459446 valid_loss 0.01698469816474244
epoch: 385 train_loss 0.008664637695377071 valid_loss 0.016976080365323772
epoch: 386 train_loss 0.00866556866482521 valid_loss 0.01696751157287508 2
```

```
epoch: 387 train_loss 0.008666512183845043 valid_loss 0.016959029925055803
epoch: 388 train_loss 0.008667473339786132 valid_loss 0.016950589740493644
epoch: 389 train_loss 0.008668453498588254 valid_loss 0.016942214271208893
epoch: 390 train_loss 0.008669446967542172 valid_loss 0.016933895418575654
epoch: 391 train_loss 0.008670452171160529 valid_loss 0.01692563099398588
epoch: 392 train_loss 0.008671478574008991 valid_loss 0.01691742689969639
epoch: 393 train_loss 0.008672520522183428 valid_loss 0.01690929460649689
epoch: 394 train_loss 0.008673575997818261 valid_loss 0.01690119137832274
epoch: 395 train_loss 0.008674651330026487 valid_loss 0.016893163249672702
epoch: 396 train_loss 0.00867574440004925 valid_loss 0.01688517606429135
epoch: 397 train_loss 0.008676840838355323 valid_loss 0.01687726778521513
epoch: 398 train_loss 0.00867796439755087 valid_loss 0.016869388097741952
epoch: 399 train_loss 0.00867910054900373 valid_loss 0.016861588309984653
epoch: 400 train_loss 0.008680246813067545 valid_loss 0.016853812104091048
epoch: 401 train_loss 0.008681410636442403 valid_loss 0.016846121929120272
epoch: 402 train_loss 0.0086825893415759 valid_loss 0.016838453997236987
epoch: 403 train_loss 0.008683782583102585 valid_loss 0.016830848029348998
epoch: 404 train_loss 0.008684988366439939 valid_loss 0.016823304091424993
epoch: 405 train_loss 0.008686212198032688 valid_loss 0.01681579153519124
epoch: 406 train_loss 0.008687445591203869 valid_loss 0.01680836364006003
epoch: 407 train_loss 0.008688696543686092 valid_loss 0.01680096089063833
epoch: 408 train_loss 0.008689958171453327 valid_loss 0.016793617303483187
epoch: 409 train_loss 0.008691236229302983 valid_loss 0.016786319265762965
epoch: 410 train_loss 0.008692528621759265 valid_loss 0.016779081150889396
epoch: 411 train_loss 0.00869383457272003 valid_loss 0.016771871181360136
epoch: 412 train_loss 0.008695153038327892 valid_loss 0.016764725748604783
epoch: 413 train_loss 0.008696484631703546 valid_loss 0.016757625764391073
epoch: 414 train_loss 0.008697827990787724 valid_loss 0.016750561562366782
epoch: 415 train_loss 0.00869918556806321 valid_loss 0.01674355985984827
epoch: 416 train_loss 0.008700556020873289 valid_loss 0.01673658745906626
epoch: 417 train_loss 0.008701943563452611 valid_loss 0.01672968777905529
epoch: 418 train_loss 0.008703334159993878 valid_loss 0.016722815487689027
epoch: 419 train_loss 0.008704750321339817 valid_loss 0.016715994310410074
epoch: 420 train_loss 0.008706164391090473 valid_loss 0.016709226315530637
epoch: 421 train_loss 0.008707602046585331 valid_loss 0.016702487781488647
epoch: 422 train_loss 0.008709049480967224 valid_loss 0.01669580499874428
epoch: 423 train_loss 0.008710503477292757 valid_loss 0.01668915736178557
epoch: 424 train_loss 0.008711976984826227 valid_loss 0.016682557548241068
epoch: 425 train_loss 0.008713459859912594 valid_loss 0.016675998802141597
epoch: 426 train_loss 0.008714954589959235 valid_loss 0.016669489617925137
epoch: 427 train_loss 0.008716457073266308 valid_loss 0.016663030566026766
epoch: 428 train_loss 0.008717979556725671 valid_loss 0.016656594218996665
epoch: 429 train_loss 0.008719507628120482 valid_loss 0.0166502156915764
```

```
epoch: 430 train_loss 0.008721048815641553 valid_loss 0.016643857432063668
epoch: 431 train_loss 0.008722599886823446 valid_loss 0.01663756602210924
epoch: 432 train_loss 0.008724165075303365 valid_loss 0.016631309331084292
epoch: 433 train_loss 0.008725739212241023 valid_loss 0.016625100389743845
epoch: 434 train_loss 0.008727326251876852 valid_loss 0.016618909729489435
epoch: 435 train_loss 0.008728923373079548 valid_loss 0.0166127656508858
epoch: 436 train_loss 0.008730528849021842 valid_loss 0.016606673621572553
epoch: 437 train_loss 0.00873214485278974 valid_loss 0.01660060230254506
epoch: 438 train_loss 0.008733776564865063 valid_loss 0.01659460748002554
epoch: 439 train_loss 0.008735417341813444 valid_loss 0.016588614174785712
epoch: 440 train_loss 0.008737067059458543 valid_loss 0.016582667567611984
epoch: 441 train_loss 0.008738725151245793 valid_loss 0.01657674591600274
epoch: 442 train_loss 0.008740397383614132 valid_loss 0.016570898052304984
epoch: 443 train_loss 0.008742075704503804 valid_loss 0.016565056905771296
epoch: 444 train_loss 0.008743769477587194 valid_loss 0.016559270144595455
epoch: 445 train_loss 0.008745464019011707 valid_loss 0.01655352090019733
epoch: 446 train_loss 0.008747175308720519 valid_loss 0.01654779745731503
epoch: 447 train_loss 0.00874889735908558 valid_loss 0.01654210757697001
epoch: 448 train_loss 0.00875062164850533 valid_loss 0.0165364621556364
epoch: 449 train_loss 0.008752363062618921 valid_loss 0.016530848325540623
epoch: 450 train_loss 0.008754110926141341 valid_loss 0.01652527506230399
epoch: 451 train_loss 0.00875586832407862 valid_loss 0.016519728112810602
epoch: 452 train_loss 0.008757634806291511 valid_loss 0.016514213010668754
epoch: 453 train_loss 0.008759409305639565 valid_loss 0.01650875942238296
epoch: 454 train_loss 0.008761192408079902 valid_loss 0.01650330692064017
epoch: 455 train_loss 0.008762986178044229 valid_loss 0.016497900624138615
epoch: 456 train_loss 0.008764785248786211 valid_loss 0.01649252619051064
epoch: 457 train_loss 0.008766594141100844 valid_loss 0.016487179595666628
epoch: 458 train_loss 0.008768416529831787 valid_loss 0.016481878887861966
epoch: 459 train_loss 0.008770243796364714 valid_loss 0.016476609266828746
epoch: 460 train_loss 0.008772077353205532 valid_loss 0.016471366009985406
epoch: 461 train_loss 0.008773916525145371 valid_loss 0.01646614750303949
epoch: 462 train_loss 0.00877576752876242 valid_loss 0.016460972395725547
epoch: 463 train_loss 0.008777627853366237 valid_loss 0.01645582801429555
epoch: 464 train_loss 0.008779493432181576 valid_loss 0.016450721444562076
epoch: 465 train_loss 0.008781366822465013 valid_loss 0.01644561814221864
epoch: 466 train_loss 0.008783250278793276 valid_loss 0.01644056347431615
epoch: 467 train_loss 0.008785136707592756 valid_loss 0.016435535756560662
epoch: 468 train_loss 0.00878703525910775 valid_loss 0.016430542843105893
epoch: 469 train_loss 0.008788933515703926 valid_loss 0.016425566491670908
epoch: 470 train_loss 0.008790844647834698 valid_loss 0.016420629015192388
epoch: 471 train_loss 0.008792760502547025 valid_loss 0.01641572586571177
epoch: 472 train_loss 0.008794685045722872 valid_loss 0.016410836992630115
```

```
epoch: 473 train_loss 0.008796612549728404 valid_loss 0.016405991138890387
epoch: 474 train_loss 0.008798550112017741 valid_loss 0.01640115197515115
epoch: 475 train_loss 0.008800490258727223 valid_loss 0.016396358011600873
epoch: 476 train_loss 0.00880244205861042 valid_loss 0.016391586244571953
epoch: 477 train_loss 0.008804393582977355 valid_loss 0.01638684921975558
epoch: 478 train_loss 0.0088063588210692 valid_loss 0.016382142148601512
epoch: 479 train_loss 0.00880832716356963 valid_loss 0.01637743345539396
epoch: 480 train_loss 0.008810298711371919 valid_loss 0.016372764700402817
epoch: 481 train_loss 0.008812276475752393 valid_loss 0.01636813295384248
epoch: 482 train_loss 0.008814258260341983 valid_loss 0.016363513683124133
epoch: 483 train_loss 0.008816251426469534 valid_loss 0.016358931843812267
epoch: 484 train_loss 0.008818245267805953 valid_loss 0.0163543637143448
epoch: 485 train_loss 0.008820248496097824 valid_loss 0.016349833252994963
epoch: 486 train_loss 0.008822250874557843 valid_loss 0.016345317184459418
epoch: 487 train_loss 0.008824263307421159 valid_loss 0.016340836820503076
epoch: 488 train_loss 0.008826279364681493 valid_loss 0.016336381155997514
epoch: 489 train_loss 0.00882830061406518 valid_loss 0.016331931572252264
epoch: 490 train_loss 0.008830330707132817 valid_loss 0.016327515093144028
epoch: 491 train_loss 0.00883235955843702 valid_loss 0.016323122735290477
epoch: 492 train_loss 0.008834396811046948 valid_loss 0.016318751708604397
epoch: 493 train_loss 0.008836436931354304 valid_loss 0.016314413119107484
epoch: 494 train_loss 0.008838481630664318 valid_loss 0.01631009115371853
epoch: 495 train_loss 0.008840532728936524 valid_loss 0.016305802192073317
epoch: 496 train_loss 0.008842585678212344 valid_loss 0.016301519835057357
epoch: 497 train_loss 0.008844642550684512 valid_loss 0.016297263489104808
epoch: 498 train_loss 0.008846709795761853 valid_loss 0.01629304449694852
epoch: 499 train_loss 0.008848777851865938 valid_loss 0.016288826001497607
epoch: 500 train_loss 0.00885084610975658 valid_loss 0.016284648635579893
epoch: 501 train_loss 0.008852925718141098 valid_loss 0.01628049419571956
epoch: 502 train_loss 0.008855006929176549 valid_loss 0.01627634217729792
epoch: 503 train_loss 0.008857091807294637 valid_loss 0.016272223317840447
epoch: 504 train_loss 0.008859177275250355 valid_loss 0.01626812726802503
epoch: 505 train_loss 0.008861269472011676 valid_loss 0.016264052041030177
epoch: 506 train_loss 0.008863368207433572 valid_loss 0.01626000030276676
epoch: 507 train_loss 0.00886546860371406 valid_loss 0.01625596423788617
epoch: 508 train_loss 0.008867570602645477 valid_loss 0.01625195684997986
epoch: 509 train_loss 0.008869682811200619 valid_loss 0.016247964871581644
epoch: 510 train_loss 0.008871792020120969 valid_loss 0.01624398654482017
epoch: 511 train_loss 0.008873910976884265 valid_loss 0.016240041493438183
epoch: 512 train_loss 0.008876031357795 valid_loss 0.016236114183751246
epoch: 513 train_loss 0.008878154567598055 valid_loss 0.0162322058264787
epoch: 514 train_loss 0.008880281444483747 valid_loss 0.016228316340129823
epoch: 515 train_loss 0.008882413645430158 valid_loss 0.016224454246306173
```

```
epoch: 516 train_loss 0.008884549086603025 valid_loss 0.01622059924605613
epoch: 517 train_loss 0.008886681535902123 valid_loss 0.0162167812542369
epoch: 518 train_loss 0.008888824800184618 valid_loss 0.016212960092040398
epoch: 519 train_loss 0.008890972887941946 valid_loss 0.016209166299086065
epoch: 520 train_loss 0.008893116326847424 valid_loss 0.01620539844734594
epoch: 521 train_loss 0.008895274492291112 valid_loss 0.016201649625630428
epoch: 522 train_loss 0.008897425831916432 valid_loss 0.016197919313951085
epoch: 523 train_loss 0.00889958415258055 valid_loss 0.016194205242209138
epoch: 524 train_loss 0.008901745748395721 valid_loss 0.016190518023601424
epoch: 525 train_loss 0.008903909431925665 valid_loss 0.016186836652923375
epoch: 526 train_loss 0.008906079506656776 valid_loss 0.016183180878094085
epoch: 527 train_loss 0.008908251665222149 valid_loss 0.01617954884034892
epoch: 528 train_loss 0.008910429671717186 valid_loss 0.01617593615083024
epoch: 529 train_loss 0.008912603879192222 valid_loss 0.01617232864179338
epoch: 530 train_loss 0.008914786779011289 valid_loss 0.01616874572743351
epoch: 531 train_loss 0.008916969659427803 valid_loss 0.016165182991729428
epoch: 532 train_loss 0.00891915955968822 valid_loss 0.016161636915057898
epoch: 533 train_loss 0.00892135053485011 valid_loss 0.016158114175777883
epoch: 534 train_loss 0.008923542441334576 valid_loss 0.016154611374561985
epoch: 535 train_loss 0.008925736043602228 valid_loss 0.016151108604390174
epoch: 536 train_loss 0.00892793651825438 valid_loss 0.016147643013391645
epoch: 537 train_loss 0.008930138129896174 valid_loss 0.01614418028621003
epoch: 538 train_loss 0.00893234348623082 valid_loss 0.0161407473264262l
epoch: 539 train_loss 0.008934553328435868 valid_loss 0.01613731721105675
epoch: 540 train_loss 0.0089367645753858 valid_loss 0.01613391706875215
epoch: 541 train_loss 0.008938976598437875 valid_loss 0.01613053473799179
epoch: 542 train_loss 0.008941195412383725 valid_loss 0.016127165767829865
epoch: 543 train_loss 0.008943414990790188 valid_loss 0.016123806145818282
epoch: 544 train_loss 0.008945634196667622 valid_loss 0.016120467292300114
epoch: 545 train_loss 0.008947863227998216 valid_loss 0.016117148000436524
epoch: 546 train_loss 0.008950091646208117 valid_loss 0.016113857769717774
epoch: 547 train_loss 0.008952321577817201 valid_loss 0.016110563914602003
epoch: 548 train_loss 0.008954550275423875 valid_loss 0.01610730222503965
epoch: 549 train_loss 0.008956789644435048 valid_loss 0.01610404736129567
epoch: 550 train_loss 0.008959023584611713 valid_loss 0.016100810060743244
epoch: 551 train_loss 0.008961265045218169 valid_loss 0.016097581041200706
epoch: 552 train_loss 0.008963511721231044 valid_loss 0.016094384978835783
epoch: 553 train_loss 0.008965755614917725 valid_loss 0.01609118711979439
epoch: 554 train_loss 0.008968005542798588 valid_loss 0.01608801434049383
epoch: 555 train_loss 0.00897025572679316 valid_loss 0.016084866101543107
epoch: 556 train_loss 0.008972506826588263 valid_loss 0.01608171914704144
epoch: 557 train_loss 0.008974764368031175 valid_loss 0.016078606147008637
epoch: 558 train_loss 0.008977020500848691 valid_loss 0.01607548534326876
```

```
epoch: 559 train_loss 0.008979281984890501 valid_loss 0.016072393371723594
epoch: 560 train_loss 0.008981543562064568 valid_loss 0.016069320791090528
epoch: 561 train_loss 0.008983807409337411 valid_loss 0.01606625281274319
epoch: 562 train_loss 0.008986073887596528 valid_loss 0.01606319985197236
epoch: 563 train_loss 0.008988348173443229 valid_loss 0.016060169813378402
epoch: 564 train_loss 0.008990617690142245 valid_loss 0.016057142231147736
epoch: 565 train_loss 0.00899288741638884 valid_loss 0.016054147183119007
epoch: 566 train_loss 0.008995165520658096 valid_loss 0.016051156865432858
epoch: 567 train_loss 0.008997445250861347 valid_loss 0.01604818079310159
epoch: 568 train_loss 0.008999726048205047 valid_loss 0.016045220526090514
epoch: 569 train_loss 0.009002005720200637 valid_loss 0.016042281225478898
epoch: 570 train_loss 0.009004293312318622 valid_loss 0.016039347272211065
epoch: 571 train_loss 0.009006583434529602 valid_loss 0.016036422348891695
epoch: 572 train_loss 0.009008872753474862 valid_loss 0.0160335327576225
epoch: 573 train_loss 0.009011162879566352 valid_loss 0.0160306470701471
epoch: 574 train_loss 0.009013455927682419 valid_loss 0.016027772236460198
epoch: 575 train_loss 0.009015750462034096 valid_loss 0.016024910868145524
epoch: 576 train_loss 0.009018050875359526 valid_loss 0.016022060287650675
epoch: 577 train_loss 0.00902034529717639 valid_loss 0.016019238686809937
epoch: 578 train_loss 0.009022648097015917 valid_loss 0.01601641777670011
epoch: 579 train_loss 0.009024950116872788 valid_loss 0.016013612446840852
epoch: 580 train_loss 0.00902725342893973 valid_loss 0.016010822782603403
epoch: 581 train_loss 0.009029562658785533 valid_loss 0.016008049928738426
epoch: 582 train_loss 0.009031873460238178 valid_loss 0.016005295662519833
epoch: 583 train_loss 0.009034182864706964 valid_loss 0.016002541260483363
epoch: 584 train_loss 0.009036494915684064 valid_loss 0.015999803874486438
epoch: 585 train_loss 0.009038811169254283 valid_loss 0.015997083477365472
epoch: 586 train_loss 0.00904112522645543 valid_loss 0.01599436595182245
epoch: 587 train_loss 0.009043444650402913 valid_loss 0.01599168110017975
epoch: 588 train_loss 0.009045766654890031 valid_loss 0.01598899046269556
epoch: 589 train_loss 0.00904808686658119 valid_loss 0.015986316134997954
epoch: 590 train_loss 0.009050408583910515 valid_loss 0.015983676557273913
epoch: 591 train_loss 0.00905273538470889 valid_loss 0.015981026676793894
epoch: 592 train_loss 0.009055059344973415 valid_loss 0.015978383808396758
epoch: 593 train_loss 0.009057387663051485 valid_loss 0.015975767482692995
epoch: 594 train_loss 0.009059718899273624 valid_loss 0.015973160800058395
epoch: 595 train_loss 0.00906205178471282 valid_loss 0.015970568626653404
epoch: 596 train_loss 0.009064385927437494 valid_loss 0.015967978013213723
epoch: 597 train_loss 0.009066721870719144 valid_loss 0.015965415351092815
epoch: 598 train_loss 0.009069057903252542 valid_loss 0.015962864186925194
epoch: 599 train_loss 0.009071395189190904 valid_loss 0.015960313585431627
epoch: 600 train_loss 0.009073736898911496 valid_loss 0.015957778152854492
epoch: 601 train_loss 0.009076076225998502 valid_loss 0.01595527115665997
```

```
epoch: 602 train_loss 0.009078422968741506 valid_loss 0.015952750132419168
epoch: 603 train_loss 0.009080767584964632 valid_loss 0.0159502556353497
epoch: 604 train_loss 0.009083114475167046 valid_loss 0.015947781080224863
epoch: 605 train_loss 0.009085463713078449 valid_loss 0.01594531489148115
epoch: 606 train_loss 0.00908780643561234 valid_loss 0.01594284571086367
epoch: 607 train_loss 0.009090161874579887 valid_loss 0.015940405521541835
epoch: 608 train_loss 0.009092514674800137 valid_loss 0.015937971766106784
epoch: 609 train_loss 0.009094871782387296 valid_loss 0.015935533893449854
epoch: 610 train_loss 0.009097229724284261 valid_loss 0.01593312305243065
epoch: 611 train_loss 0.009099581092596055 valid_loss 0.015930718874248365
epoch: 612 train_loss 0.009101937955711037 valid_loss 0.015928330090052137
epoch: 613 train_loss 0.009104300282585125 valid_loss 0.01592594956746325
epoch: 614 train_loss 0.009106660925317556 valid_loss 0.015923587360884995
epoch: 615 train_loss 0.009109023997249702 valid_loss 0.015921230881940574
epoch: 616 train_loss 0.009111387057540317 valid_loss 0.01591887849305446
epoch: 617 train_loss 0.009113752857471506 valid_loss 0.015916542053067435
epoch: 618 train_loss 0.009116117846375952 valid_loss 0.015914222217785816
epoch: 619 train_loss 0.009118488299039502 valid_loss 0.015911909254888692
epoch: 620 train_loss 0.009120859128112594 valid_loss 0.015909604367334396
epoch: 621 train_loss 0.00912322533937792 valid_loss 0.015907321501678475
epoch: 622 train_loss 0.009125594057453175 valid_loss 0.01590503801126033
epoch: 623 train_loss 0.009127970373568436 valid_loss 0.015902773182218274
epoch: 624 train_loss 0.009130341520843407 valid_loss 0.015900509296140324
epoch: 625 train_loss 0.009132716455496848 valid_loss 0.01589826159955313
epoch: 626 train_loss 0.009135092154610901 valid_loss 0.01589601890494426
epoch: 627 train_loss 0.009137466456741095 valid_loss 0.015893812062374005
epoch: 628 train_loss 0.009139844340582688 valid_loss 0.015891578696512927
epoch: 629 train_loss 0.009142220435508837 valid_loss 0.015889380883891135
epoch: 630 train_loss 0.009144596829234313 valid_loss 0.015887187987876434
epoch: 631 train_loss 0.009146980370860546 valid_loss 0.015884998044930397
epoch: 632 train_loss 0.00914935905408735 valid_loss 0.01588281865697354
epoch: 633 train_loss 0.009151742463776221 valid_loss 0.01588065338631471
epoch: 634 train_loss 0.00915412042910854 valid_loss 0.01587850898116206
epoch: 635 train_loss 0.009156504673107217 valid_loss 0.015876366811183592
epoch: 636 train_loss 0.00915888644134005 valid_loss 0.015874228564401467
epoch: 637 train_loss 0.009161274325257788 valid_loss 0.01587211002673333
epoch: 638 train_loss 0.009163653520712007 valid_loss 0.01586998792287583
epoch: 639 train_loss 0.009166041307616979 valid_loss 0.015867872384842486
epoch: 640 train_loss 0.009168427728582174 valid_loss 0.015865791387235127
epoch: 641 train_loss 0.00917081266331176 valid_loss 0.015863701708925267
epoch: 642 train_loss 0.009173199689636627 valid_loss 0.01586163081228733
epoch: 643 train_loss 0.009175587038043886 valid_loss 0.01585957083928709
epoch: 644 train_loss 0.009177969074031959 valid_loss 0.015857518153886
```

```
epoch: 645 train_loss 0.009180360578466207 valid_loss 0.015855453291442244
epoch: 646 train_loss 0.00918274640571326 valid_loss 0.015853422476599613
epoch: 647 train_loss 0.009185132985779394 valid_loss 0.015851398060719172
epoch: 648 train_loss 0.00918752426126351 valid_loss 0.015849374793469907
epoch: 649 train_loss 0.009189914205732445 valid_loss 0.01584737248485908
epoch: 650 train_loss 0.00919230409199372 valid_loss 0.015845375896121065
epoch: 651 train_loss 0.0091946937997515 valid_loss 0.015843388139425468
epoch: 652 train_loss 0.009197081691430261 valid_loss 0.015841403409528235
epoch: 653 train_loss 0.009199475244774174 valid_loss 0.015839431636656322
epoch: 654 train_loss 0.009201864145385723 valid_loss 0.015837479817370572
epoch: 655 train_loss 0.009204257791861892 valid_loss 0.015835527059001226
epoch: 656 train_loss 0.009206647635437548 valid_loss 0.015833578685609005
epoch: 657 train_loss 0.009209039687023808 valid_loss 0.015831641336747755
epoch: 658 train_loss 0.009211430768482387 valid_loss 0.015829705722474805
epoch: 659 train_loss 0.00921382480300963 valid_loss 0.01582779873861
epoch: 660 train_loss 0.009216215911631782 valid_loss 0.015825882876136652
epoch: 661 train_loss 0.009218607322933774 valid_loss 0.015823990277325114
epoch: 662 train_loss 0.009221005078870803 valid_loss 0.015822098792220154
epoch: 663 train_loss 0.009223399229813366 valid_loss 0.015820211498066783
epoch: 664 train_loss 0.009225793493290743 valid_loss 0.015818338577325144
epoch: 665 train_loss 0.009228188272876045 valid_loss 0.015816483995877206
epoch: 666 train_loss 0.009230582567397505 valid_loss 0.015814622010414798
epoch: 667 train_loss 0.009232975884030263 valid_loss 0.015812783098469178
epoch: 668 train_loss 0.00923536914633587 valid_loss 0.01581093668937683
epoch: 669 train_loss 0.009237765974830836 valid_loss 0.01580911180159698
epoch: 670 train_loss 0.009240159768766413 valid_loss 0.015807291011636457
epoch: 671 train_loss 0.009242558056333413 valid_loss 0.015805475514692566
epoch: 672 train_loss 0.009244953681870054 valid_loss 0.01580367157779013
epoch: 673 train_loss 0.0092473510458755 valid_loss 0.0158018829068169
epoch: 674 train_loss 0.009249748142125706 valid_loss 0.01580009595102941
epoch: 675 train_loss 0.009252145486728598 valid_loss 0.015798299965293458
epoch: 676 train_loss 0.00925453931170826 valid_loss 0.01579652769335856
epoch: 677 train_loss 0.009256937067645292 valid_loss 0.015794769030374786
epoch: 678 train_loss 0.00925933209558328 valid_loss 0.015793017100077123
epoch: 679 train_loss 0.009261732963689913 valid_loss 0.015791267486444365
epoch: 680 train_loss 0.009264129609800876 valid_loss 0.015789529545387874
epoch: 681 train_loss 0.009266529278829694 valid_loss 0.015787800626518824
epoch: 682 train_loss 0.00926892621597896 valid_loss 0.015786073457760116
epoch: 683 train_loss 0.009271325764711947 valid_loss 0.015784359618555753
epoch: 684 train_loss 0.00927372333050395 valid_loss 0.015782653466643145
epoch: 685 train_loss 0.00927612448964889 valid_loss 0.015780961754110954
epoch: 686 train_loss 0.009278520673979074 valid_loss 0.015779262404733648
epoch: 687 train_loss 0.009280921468356003 valid_loss 0.015777579702747364
```

```
epoch: 688 train_loss 0.009283317536270867 valid_loss 0.015775908114543807
epoch: 689 train_loss 0.009285714768338948 valid_loss 0.01577422977037107
epoch: 690 train_loss 0.009288117584461967 valid_loss 0.015772571798879653
epoch: 691 train_loss 0.009290516768426944 valid_loss 0.01577090973344942
epoch: 692 train_loss 0.009292919786336522 valid_loss 0.01576927174658825
epoch: 693 train_loss 0.009295318376583357 valid_loss 0.01576763648384561
epoch: 694 train_loss 0.009297718352172524 valid_loss 0.01576600078260526
epoch: 695 train_loss 0.00930011427262798 valid_loss 0.015764390192149827
epoch: 696 train_loss 0.009302514442242681 valid_loss 0.015762762531327704
epoch: 697 train_loss 0.009304914309177547 valid_loss 0.015761159566075853
epoch: 698 train_loss 0.009307318355422468 valid_loss 0.015759562929936997
epoch: 699 train_loss 0.0093097170892482 valid_loss 0.01575796415951724
epoch: 700 train_loss 0.009312119703584662 valid_loss 0.015756376537804803
epoch: 701 train_loss 0.009314514727642139 valid_loss 0.01575480113970116
epoch: 702 train_loss 0.009316916476624708 valid_loss 0.015753229109880824
epoch: 703 train_loss 0.009319315835212668 valid_loss 0.015751667666093757
epoch: 704 train_loss 0.009321717316439996 valid_loss 0.015750117774587125
epoch: 705 train_loss 0.009324116636222849 valid_loss 0.015748553222510963
epoch: 706 train_loss 0.009326516056898982 valid_loss 0.015747011619775247
epoch: 707 train_loss 0.009328916657250375 valid_loss 0.015745479473844172
epoch: 708 train_loss 0.009331319694562504 valid_loss 0.015743943175766616
epoch: 709 train_loss 0.009333721377576392 valid_loss 0.015742425279070934
epoch: 710 train_loss 0.009336116362828762 valid_loss 0.015740917782144
epoch: 711 train_loss 0.009338519357455274 valid_loss 0.01573940593516454
epoch: 712 train_loss 0.00934091991900156 valid_loss 0.01573790287366137
epoch: 713 train_loss 0.00934331826865673 valid_loss 0.01573640896240249
epoch: 714 train_loss 0.00934572321906065 valid_loss 0.015734924321683743
epoch: 715 train_loss 0.009348125359974802 valid_loss 0.01573344339461376
epoch: 716 train_loss 0.009350521555946519 valid_loss 0.01573196833099549
epoch: 717 train_loss 0.009352926615004737 valid_loss 0.01573049684908862
epoch: 718 train_loss 0.009355320121782522 valid_loss 0.015729040016109745
epoch: 719 train_loss 0.009357722231652588 valid_loss 0.015727587366321436
epoch: 720 train_loss 0.009360122358581672 valid_loss 0.01572613458847627
epoch: 721 train_loss 0.009362521278671921 valid_loss 0.01572470018872991
epoch: 722 train_loss 0.009364922779301803 valid_loss 0.01572327169512088
epoch: 723 train_loss 0.009367322033115973 valid_loss 0.01572184239436562
epoch: 724 train_loss 0.009369717868200194 valid_loss 0.01572041727292041
epoch: 725 train_loss 0.00937212307471782 valid_loss 0.01571901054897656
epoch: 726 train_loss 0.009374522080179304 valid_loss 0.015717608372991285
epoch: 727 train_loss 0.009376916075901439 valid_loss 0.015716197283472864
epoch: 728 train_loss 0.009379316590881596 valid_loss 0.015714818603980045
epoch: 729 train_loss 0.009381716772137831 valid_loss 0.01571342154250791
epoch: 730 train_loss 0.009384116429525117 valid_loss 0.01571203947144871
```

```
epoch: 731 train_loss 0.009386510401964188 valid_loss 0.015710656558318683
epoch: 732 train_loss 0.009388912073336541 valid_loss 0.015709282473350565
epoch: 733 train_loss 0.009391312021762133 valid_loss 0.015707935528674475
epoch: 734 train_loss 0.009393707860726862 valid_loss 0.01570656808714072
epoch: 735 train_loss 0.009396101379146178 valid_loss 0.01570522072336947
epoch: 736 train_loss 0.009398502887537082 valid_loss 0.015703877760097386
epoch: 737 train_loss 0.009400905373816689 valid_loss 0.015702551502423983
epoch: 738 train_loss 0.00940329699854677 valid_loss 0.015701210494929303
epoch: 739 train_loss 0.009405693442871173 valid_loss 0.0156998906400986
epoch: 740 train_loss 0.009408092188338438 valid_loss 0.01569856172039484
epoch: 741 train_loss 0.00941049117827788 valid_loss 0.01569725084506596
epoch: 742 train_loss 0.009412886582625409 valid_loss 0.0156959469934615
epoch: 743 train_loss 0.009415285906288772 valid_loss 0.015694643432895342
epoch: 744 train_loss 0.009417682761947314 valid_loss 0.015693358165056754
epoch: 745 train_loss 0.00942007255119582 valid_loss 0.015692060564955077
epoch: 746 train_loss 0.009422475557463865 valid_loss 0.015690781455487014
epoch: 747 train_loss 0.009424868156202137 valid_loss 0.015689502097666265
epoch: 748 train_loss 0.009427262753403436 valid_loss 0.015688231932775427
epoch: 749 train_loss 0.009429656143765897 valid_loss 0.015686970145907253
epoch: 750 train_loss 0.009432052844204009 valid_loss 0.015685705246869474
epoch: 751 train_loss 0.00943444633934026 valid_loss 0.015684459013088296
epoch: 752 train_loss 0.009436842954407136 valid_loss 0.015683210377270978
epoch: 753 train_loss 0.009439236290442446 valid_loss 0.015681965738379707
epoch: 754 train_loss 0.009441629498420904 valid_loss 0.01568072473552699
epoch: 755 train_loss 0.009444021460755419 valid_loss 0.015679500019177794
epoch: 756 train_loss 0.009446417974929015 valid_loss 0.015678270685020833
epoch: 757 train_loss 0.009448807158817848 valid_loss 0.01567706037700797
epoch: 758 train_loss 0.009451201554232588 valid_loss 0.015675834054127334
epoch: 759 train_loss 0.009453593788202852 valid_loss 0.015674638748168945
epoch: 760 train_loss 0.009455987190206846 valid_loss 0.015673432394396513
epoch: 761 train_loss 0.009458378776131818 valid_loss 0.015672232140786946
epoch: 762 train_loss 0.009460769838187843 valid_loss 0.015671039702525983
epoch: 763 train_loss 0.00946315829254066 valid_loss 0.015669855921684454
epoch: 764 train_loss 0.009465548772520075 valid_loss 0.01566867355334883
epoch: 765 train_loss 0.009467944324327012 valid_loss 0.01566750331937025
epoch: 766 train_loss 0.009470330791858335 valid_loss 0.015666330171128114
epoch: 767 train_loss 0.00947272350701193 valid_loss 0.015665168509197733
epoch: 768 train_loss 0.009475110610947014 valid_loss 0.01566401052211101
epoch: 769 train_loss 0.009477496290734659 valid_loss 0.015662865818012506
epoch: 770 train_loss 0.009479887271299958 valid_loss 0.015661702103291947
epoch: 771 train_loss 0.009482275931319843 valid_loss 0.01566056830342859
epoch: 772 train_loss 0.009484661277383566 valid_loss 0.015659434204765905
epoch: 773 train_loss 0.009487051834973196 valid_loss 0.015658291491369405
```

```
epoch: 774 train_loss 0.009489439288154244 valid_loss 0.015657170431222767
epoch: 775 train_loss 0.009491826058365405 valid_loss 0.015656055494522054
epoch: 776 train_loss 0.009494212576343367 valid_loss 0.015654945090257874
epoch: 777 train_loss 0.009496599261183292 valid_loss 0.015653830277733503
epoch: 778 train_loss 0.009498983842786402 valid_loss 0.015652723020563523
epoch: 779 train_loss 0.009501370481060197 valid_loss 0.01565162875922397
epoch: 780 train_loss 0.0095037573754477 valid_loss 0.015650524699594826
epoch: 781 train_loss 0.009506137820426374 valid_loss 0.015649438730906694
epoch: 782 train_loss 0.009508523426484316 valid_loss 0.015648360080861796
epoch: 783 train_loss 0.009510906657669693 valid_loss 0.015647279882493117
epoch: 784 train_loss 0.009513291926123203 valid_loss 0.015646200782308977
epoch: 785 train_loss 0.009515674769257505 valid_loss 0.015645118053847305
epoch: 786 train_loss 0.009518054550668845 valid_loss 0.01564406822047507
epoch: 787 train_loss 0.0095204374481303 valid_loss 0.015643010986968874
epoch: 788 train_loss 0.009522819080545256 valid_loss 0.01564195294631645
epoch: 789 train_loss 0.009525200662513575 valid_loss 0.0156409087862509
epoch: 790 train_loss 0.009527580917347223 valid_loss 0.015639861929230392
epoch: 791 train_loss 0.009529959612215559 valid_loss 0.015638824206932137
epoch: 792 train_loss 0.009532343343986819 valid_loss 0.015637788471455374
epoch: 793 train_loss 0.009534723990752052 valid_loss 0.01563676034566015
epoch: 794 train_loss 0.009537101967725902 valid_loss 0.015635733384018143
epoch: 795 train_loss 0.009539479964102308 valid_loss 0.01563471267775943
epoch: 796 train_loss 0.009541860432364047 valid_loss 0.015633686166256665
epoch: 797 train_loss 0.009544232235445331 valid_loss 0.015632684672406565
epoch: 798 train_loss 0.00954661612631753 valid_loss 0.01563167912342275
epoch: 799 train_loss 0.009548988379538059 valid_loss 0.015630677451069157
epoch: 800 train_loss 0.00955136576279377 valid_loss 0.015629674432178338
epoch: 801 train_loss 0.009553740631478529 valid_loss 0.015628699337442716
epoch: 802 train_loss 0.009556117032965024 valid_loss 0.01562770624877885
epoch: 803 train_loss 0.009558491327334195 valid_loss 0.015626735942593464
epoch: 804 train_loss 0.009560866238704573 valid_loss 0.015625752815200636
epoch: 805 train_loss 0.00956323965995883 valid_loss 0.015624776393330346
epoch: 806 train_loss 0.009565613535232841 valid_loss 0.015623812105817099
epoch: 807 train_loss 0.009567986913801481 valid_loss 0.015622850829580178
epoch: 808 train_loss 0.009570357059904685 valid_loss 0.015621889530060192
epoch: 809 train_loss 0.009572731707400332 valid_loss 0.015620939895355453
epoch: 810 train_loss 0.009575104364193976 valid_loss 0.0156199880487596
epoch: 811 train_loss 0.00957747536400954 valid_loss 0.015619048608156542
epoch: 812 train_loss 0.009579844419689228 valid_loss 0.015618102139948556
epoch: 813 train_loss 0.009582213778048753 valid_loss 0.01561716467452546
epoch: 814 train_loss 0.009584584013403704 valid_loss 0.015616245036168645
epoch: 815 train_loss 0.009586955366345743 valid_loss 0.015615316689945757
epoch: 816 train_loss 0.009589319955557585 valid_loss 0.015614396364738544
```

```
epoch: 817 train_loss 0.009591690400460115 valid_loss 0.015613470707709591
epoch: 818 train_loss 0.009594057958262662 valid_loss 0.015612553560640664
epoch: 819 train_loss 0.009596425698449214 valid_loss 0.015611650907279303
epoch: 820 train_loss 0.00959879239477838 valid_loss 0.015610747807659208
epoch: 821 train_loss 0.009601159634379049 valid_loss 0.015609840431716293
epoch: 822 train_loss 0.009603520719489703 valid_loss 0.015608946909196675
epoch: 823 train_loss 0.009605884831398726 valid_loss 0.015608056945105394
epoch: 824 train_loss 0.009608252230100334 valid_loss 0.01560717741958797
epoch: 825 train_loss 0.009610614203847945 valid_loss 0.015606287893994401
epoch: 826 train_loss 0.009612972254399211 valid_loss 0.01560540945113947
epoch: 827 train_loss 0.009615340289504578 valid_loss 0.015604533833296349
epoch: 828 train_loss 0.009617702659064283 valid_loss 0.015603677420100819
epoch: 829 train_loss 0.00962006520324697 valid_loss 0.015602807277658334
epoch: 830 train_loss 0.009622425065996747 valid_loss 0.015601942164357752
epoch: 831 train_loss 0.009624788083601742 valid_loss 0.015601089608389884
epoch: 832 train_loss 0.00962714475269119 valid_loss 0.015600233470710616
epoch: 833 train_loss 0.009629506497488668 valid_loss 0.015599386964458973
epoch: 834 train_loss 0.009631864714901895 valid_loss 0.015598534462818255
epoch: 835 train_loss 0.009634224030499657 valid_loss 0.015597709858169159
epoch: 836 train_loss 0.009636578681723525 valid_loss 0.015596860519144684
epoch: 837 train_loss 0.009638936390789847 valid_loss 0.015596026372319709
epoch: 838 train_loss 0.009641293319873512 valid_loss 0.015595209451081853
epoch: 839 train_loss 0.009643651975784451 valid_loss 0.015594375238288194
epoch: 840 train_loss 0.00964600567240268 valid_loss 0.01559356505361696
epoch: 841 train_loss 0.00964835963677615 valid_loss 0.015592748826990525
epoch: 842 train_loss 0.00965071755927056 valid_loss 0.015591926810642084
epoch: 843 train_loss 0.009653066587634384 valid_loss 0.015591124817728997
epoch: 844 train_loss 0.009655421456166853 valid_loss 0.0155903271981515
epoch: 845 train_loss 0.009657774011914928 valid_loss 0.015589525674780209
epoch: 846 train_loss 0.009660121561804165 valid_loss 0.015588737562453995
epoch: 847 train_loss 0.009662476321682333 valid_loss 0.01558794843343397
epoch: 848 train_loss 0.009664828636838744 valid_loss 0.015587150189094245
epoch: 849 train_loss 0.009667177983404447 valid_loss 0.015586370058978597
epoch: 850 train_loss 0.009669527361014237 valid_loss 0.015585594236229856
epoch: 851 train_loss 0.009671876090578735 valid_loss 0.015584817963341872
epoch: 852 train_loss 0.00967422454074646 valid_loss 0.01558404842702051
epoch: 853 train_loss 0.009676573351801684 valid_loss 0.01558328108706822
epoch: 854 train_loss 0.009678919000240664 valid_loss 0.01558251358413448
epoch: 855 train_loss 0.009681265234636765 valid_loss 0.015581748165034999
epoch: 856 train_loss 0.009683611915291598 valid_loss 0.015580999439892669
epoch: 857 train_loss 0.00968595949234441 valid_loss 0.015580244455486536
epoch: 858 train_loss 0.009688302436067412 valid_loss 0.015579497162252664
epoch: 859 train_loss 0.009690649252540122 valid_loss 0.015578758887325723
```

```
epoch: 860 train_loss 0.009692992184621592 valid_loss 0.01557802374009043
epoch: 861 train_loss 0.009695334554029008 valid_loss 0.01557728114227454
epoch: 862 train_loss 0.009697675153923531 valid_loss 0.015576537942979484
epoch: 863 train_loss 0.00970001871076723 valid_loss 0.015575812202102194
epoch: 864 train_loss 0.009702359729756912 valid_loss 0.015575083321891725
epoch: 865 train_loss 0.009704695944674314 valid_loss 0.015574364709512642
epoch: 866 train_loss 0.009707041635798911 valid_loss 0.015573643613606691
epoch: 867 train_loss 0.009709377858477334 valid_loss 0.01557293050379182
epoch: 868 train_loss 0.009711718458371857 valid_loss 0.015572222787886858
epoch: 869 train_loss 0.009714055488196512 valid_loss 0.015571518009528518
epoch: 870 train_loss 0.00971638848229001 valid_loss 0.01557081398786977
epoch: 871 train_loss 0.009718727689081183 valid_loss 0.01557011673382173
epoch: 872 train_loss 0.009721066989004611 valid_loss 0.01556942299939692
epoch: 873 train_loss 0.00972339982787768 valid_loss 0.01556873124015207
epoch: 874 train_loss 0.009725730097852648 valid_loss 0.015568040051342298
epoch: 875 train_loss 0.00972807058521236 valid_loss 0.015567348439556856
epoch: 876 train_loss 0.009730403032153844 valid_loss 0.015566671647441884
epoch: 877 train_loss 0.00973273445852101 valid_loss 0.015565990024091056
epoch: 878 train_loss 0.009735064771181594 valid_loss 0.015565315897886952
epoch: 879 train_loss 0.009737399441655725 valid_loss 0.01556464423192665
epoch: 880 train_loss 0.009739730631311734 valid_loss 0.015563979772074769
epoch: 881 train_loss 0.009742061879175404 valid_loss 0.015563316484137129
epoch: 882 train_loss 0.009744390173970411 valid_loss 0.015562665509060025
epoch: 883 train_loss 0.009746715185853343 valid_loss 0.01556200347840786
epoch: 884 train_loss 0.009749049390666186 valid_loss 0.015561339142732323
epoch: 885 train_loss 0.00975137489925449 valid_loss 0.015560706801867734
epoch: 886 train_loss 0.009753698180429638 valid_loss 0.015560047673837592
epoch: 887 train_loss 0.009756028504731755 valid_loss 0.015559406027508278
epoch: 888 train_loss 0.009758355437467496 valid_loss 0.015558773543064792
epoch: 889 train_loss 0.009760677969704071 valid_loss 0.015558132079119484
epoch: 890 train_loss 0.009763001716540506 valid_loss 0.015557502318794528
epoch: 891 train_loss 0.009765328893748423 valid_loss 0.015556868421845138
epoch: 892 train_loss 0.009767649974673987 valid_loss 0.015556245130331567
epoch: 893 train_loss 0.00976997612354656 valid_loss 0.015555619886921097
epoch: 894 train_loss 0.009772295275858293 valid_loss 0.015554996513916801
epoch: 895 train_loss 0.0097746193020915 valid_loss 0.01555439761141315
epoch: 896 train_loss 0.009776933987935384 valid_loss 0.01555378307821229
epoch: 897 train_loss 0.009779257009116311 valid_loss 0.015553170667650799
epoch: 898 train_loss 0.00978157587427025 valid_loss 0.015552562048348288
epoch: 899 train_loss 0.00978389208127434 valid_loss 0.015551957957601796
epoch: 900 train_loss 0.009786212366695206 valid_loss 0.015551357630950708
epoch: 901 train_loss 0.009788530059934904 valid_loss 0.015550764378470678
epoch: 902 train_loss 0.009790847295274336 valid_loss 0.015550171118229627
```

```
epoch: 903 train_loss 0.009793163956298182 valid_loss 0.015549582964740693
epoch: 904 train_loss 0.009795475277739267 valid_loss 0.01554899481125176
epoch: 905 train_loss 0.009797791282956799 valid_loss 0.015548415567415456
epoch: 906 train_loss 0.00980010221634681 valid_loss 0.015547842986416071
epoch: 907 train_loss 0.009802418640659501 valid_loss 0.015547260145346324
epoch: 908 train_loss 0.00980473291904976 valid_loss 0.015546687416887531
epoch: 909 train_loss 0.009807037403030941 valid_loss 0.015546124273290237
epoch: 910 train_loss 0.00980934837134555 valid_loss 0.015545558619002501
epoch: 911 train_loss 0.009811660030391068 valid_loss 0.015544987089621523
epoch: 912 train_loss 0.00981397161182637 valid_loss 0.015544419999544819
epoch: 913 train_loss 0.009816280112136155 valid_loss 0.01554387299499164
epoch: 914 train_loss 0.009818589683466902 valid_loss 0.015543317891812573
epoch: 915 train_loss 0.009820896107703448 valid_loss 0.015542767383158207
epoch: 916 train_loss 0.009823202609550209 valid_loss 0.015542220456215243
epoch: 917 train_loss 0.00982551008152465 valid_loss 0.015541674534324557
epoch: 918 train_loss 0.009827816435912002 valid_loss 0.015541131693559388
epoch: 919 train_loss 0.009830118610989303 valid_loss 0.015540589671581983
epoch: 920 train_loss 0.009832425143880148 valid_loss 0.01554006536801656
epoch: 921 train_loss 0.00983472476558139 valid_loss 0.015539532073307782
epoch: 922 train_loss 0.009837031736969948 valid_loss 0.015539001626893877
epoch: 923 train_loss 0.009839331288821996 valid_loss 0.015538479244181265
epoch: 924 train_loss 0.009841632284224033 valid_loss 0.015537953295279294
epoch: 925 train_loss 0.009843932837247849 valid_loss 0.015537436989446481
epoch: 926 train_loss 0.009846230561379343 valid_loss 0.015536910264442365
epoch: 927 train_loss 0.009848530819484343 valid_loss 0.015536398968348901
epoch: 928 train_loss 0.009850829436133305 valid_loss 0.015535895774761836
epoch: 929 train_loss 0.009853128002335627 valid_loss 0.015535382212450106
epoch: 930 train_loss 0.009855424868874252 valid_loss 0.015534883970394731
epoch: 931 train_loss 0.009857722014809648 valid_loss 0.015534376543170463
epoch: 932 train_loss 0.009860015284114827 valid_loss 0.015533876570407302
epoch: 933 train_loss 0.00986231182469055 valid_loss 0.015533388394396752
epoch: 934 train_loss 0.009864607821994772 valid_loss 0.015532897835752617
epoch: 935 train_loss 0.009866899950429798 valid_loss 0.015532400144729763
epoch: 936 train_loss 0.009869193169288337 valid_loss 0.01553191472776234
epoch: 937 train_loss 0.009871484882508715 valid_loss 0.015531436489739765
epoch: 938 train_loss 0.009873773607735833 valid_loss 0.015530944894999265
epoch: 939 train_loss 0.009876066748984159 valid_loss 0.015530469369453688
epoch: 940 train_loss 0.009878355400481572 valid_loss 0.015529993987487008
epoch: 941 train_loss 0.009880645340308547 valid_loss 0.015529531477174412
epoch: 942 train_loss 0.009882935493563613 valid_loss 0.015529060363769531
epoch: 943 train_loss 0.009885223465971649 valid_loss 0.015528581803664565
epoch: 944 train_loss 0.009887508469788978 valid_loss 0.015528118249494582
epoch: 945 train_loss 0.00988979492879783 valid_loss 0.015527668063684056
```

```
epoch: 946 train_loss 0.009892081674964477 valid_loss 0.015527200679449986
epoch: 947 train_loss 0.009894367412198336 valid_loss 0.015526753927891454
epoch: 948 train_loss 0.009896649299965551 valid_loss 0.015526299741274367
epoch: 949 train_loss 0.009898932569194585 valid_loss 0.015525841592655827
epoch: 950 train_loss 0.009901215314554672 valid_loss 0.015525399303684632
epoch: 951 train_loss 0.00990350234011809 valid_loss 0.015524961845949292
epoch: 952 train_loss 0.00990578274941072 valid_loss 0.015524511050898582
epoch: 953 train_loss 0.009908058397316684 valid_loss 0.015524074664184203
epoch: 954 train_loss 0.009910343284718692 valid_loss 0.015523643415266027
epoch: 955 train_loss 0.009912619021876404 valid_loss 0.01552321781249096
epoch: 956 train_loss 0.009914899993843089 valid_loss 0.015522781868154804
epoch: 957 train_loss 0.009917177096940576 valid_loss 0.015522354030205557
epoch: 958 train_loss 0.009919452372317512 valid_loss 0.015521931710342567
epoch: 959 train_loss 0.00992173261474818 valid_loss 0.015521504357457161
epoch: 960 train_loss 0.009924003757381191 valid_loss 0.015521083178464323
epoch: 961 train_loss 0.009926280763465911 valid_loss 0.015520668933944156
epoch: 962 train_loss 0.009928556170780211 valid_loss 0.015520253179905316
epoch: 963 train_loss 0.00993082810503741 valid_loss 0.015519846645959963
epoch: 964 train_loss 0.009933101009422293 valid_loss 0.01551943951829647
epoch: 965 train_loss 0.009935371399236222 valid_loss 0.015519036445766687
epoch: 966 train_loss 0.009937645421208193 valid_loss 0.01551863195685049
epoch: 967 train_loss 0.00993991355256488 valid_loss 0.015518228892081728
epoch: 968 train_loss 0.009942183546566714 valid_loss 0.01551783635125806
epoch: 969 train_loss 0.009944450540933758 valid_loss 0.015517434780485928
epoch: 970 train_loss 0.009946718637365848 valid_loss 0.015517050571118792
epoch: 971 train_loss 0.009948982954180488 valid_loss 0.015516661111420641
epoch: 972 train_loss 0.009951255598571152 valid_loss 0.01551626508977885
epoch: 973 train_loss 0.00995352267442892 valid_loss 0.015515882160980255
epoch: 974 train_loss 0.009955786374242355 valid_loss 0.015515507144543032
epoch: 975 train_loss 0.009958051568052422 valid_loss 0.015515119904497018
epoch: 976 train_loss 0.009960318147204817 valid_loss 0.015514750607932608
epoch: 977 train_loss 0.009962577702632794 valid_loss 0.015514363309678932
epoch: 978 train_loss 0.009964843591054281 valid_loss 0.015514004471090933
epoch: 979 train_loss 0.009967101431296517 valid_loss 0.015513627068139612
epoch: 980 train_loss 0.009969361774468173 valid_loss 0.01551326127567639
epoch: 981 train_loss 0.009971624178191026 valid_loss 0.0155129023323146
epoch: 982 train_loss 0.009973886601316432 valid_loss 0.015512536369108905
epoch: 983 train_loss 0.009976141212973744 valid_loss 0.015512177542162438
epoch: 984 train_loss 0.009978397594143948 valid_loss 0.01551181306131184
epoch: 985 train_loss 0.009980656334664672 valid_loss 0.015511462414481987
epoch: 986 train_loss 0.009982913561786214 valid_loss 0.015511114433563004
epoch: 987 train_loss 0.009985173858391741 valid_loss 0.015510760713368654
epoch: 988 train_loss 0.009987424244172871 valid_loss 0.01551042094361037
```

```
epoch: 989 train_loss 0.009989681894270082 valid_loss 0.015510073327459396
epoch: 990 train_loss 0.009991933785689374 valid_loss 0.015509735928693166
epoch: 991 train_loss 0.00999418478847171 valid_loss 0.015509401653738071
epoch: 992 train_loss 0.009996439473858724 valid_loss 0.01550905981566757
epoch: 993 train_loss 0.009998689960533132 valid_loss 0.015508722056013842
epoch: 994 train_loss 0.010000943811610341 valid_loss 0.015508398390375077
epoch: 995 train_loss 0.01000319264906769 valid_loss 0.015508066595066339
epoch: 996 train_loss 0.010005439654923976 valid_loss 0.015507725513695428
epoch: 997 train_loss 0.010007688069405656 valid_loss 0.015507417222640167
epoch: 998 train_loss 0.010009936289861798 valid_loss 0.015507096789466839
epoch: 999 train_loss 0.010012181165317694 valid_loss 0.015506772118775795
epoch: 1000 train_loss 0.010014430693505954 valid_loss 0.015506456334454318
epoch: 1001 train_loss 0.010016673795568446 valid_loss 0.015506137317667404
epoch: 1002 train_loss 0.010018921255444487 valid_loss 0.015505834121722729
epoch: 1003 train_loss 0.01002116322827836 valid_loss 0.015505511690086375
epoch: 1004 train_loss 0.010023406450636685 valid_loss 0.015505206658660124
epoch: 1005 train_loss 0.010025646743209412 valid_loss 0.015504904312547297
epoch: 1006 train_loss 0.010027889100213846 valid_loss 0.015504601411521435
epoch: 1007 train_loss 0.010030131961684674 valid_loss 0.015504299837630242
epoch: 1008 train_loss 0.01003237289066116 valid_loss 0.015504002264545609
epoch: 1009 train_loss 0.010034611580582957 valid_loss 0.015503702460167308
epoch: 1010 train_loss 0.010036852331055949 valid_loss 0.015503419745558251
epoch: 1011 train_loss 0.010039090473825733 valid_loss 0.0155031216951708
epoch: 1012 train_loss 0.010041329101659358 valid_loss 0.015502840164117515
epoch: 1013 train_loss 0.01004356392659247 valid_loss 0.015502543988016744
epoch: 1014 train_loss 0.010045800420145194 valid_loss 0.015502264893924197
epoch: 1015 train_loss 0.01004803436032186 valid_loss 0.015501983234814058
epoch: 1016 train_loss 0.010050265782047063 valid_loss 0.015501699072774499
epoch: 1017 train_loss 0.01005250345915556 valid_loss 0.015501430456060917
epoch: 1018 train_loss 0.010054736386518926 valid_loss 0.015501150267664344
epoch: 1019 train_loss 0.010056966690657039 valid_loss 0.01550088649770866
epoch: 1020 train_loss 0.010059195438710351 valid_loss 0.015500601431510101
epoch: 1021 train_loss 0.01006142731833582 valid_loss 0.015500337611107777
epoch: 1022 train_loss 0.010063660660913834 valid_loss 0.015500070232277115
epoch: 1023 train_loss 0.010065887736467024 valid_loss 0.01549981206577892
epoch: 1024 train_loss 0.010068116368105014 valid_loss 0.015499549180579683
epoch: 1025 train_loss 0.010070344712585211 valid_loss 0.015499290734684716
epoch: 1026 train_loss 0.010072569696543117 valid_loss 0.015499027465314915
epoch: 1027 train_loss 0.010074798328181108 valid_loss 0.015498772705905139
epoch: 1028 train_loss 0.010077022050973028 valid_loss 0.015498518579018612
epoch: 1029 train_loss 0.01007924813311547 valid_loss 0.015498283059181025
epoch: 1030 train_loss 0.010081474382119874 valid_loss 0.015498028936175008
epoch: 1031 train_loss 0.010083695993913958 valid_loss 0.015497772799183926
```

```
epoch: 1032 train_loss 0.010085917403921485 valid_loss 0.015497533301822842
epoch: 1033 train_loss 0.010088141270292303 valid_loss 0.015497300478940209
epoch: 1034 train_loss 0.010090361799423892 valid_loss 0.015497055808858325
epoch: 1035 train_loss 0.010092578979674726 valid_loss 0.015496814355719835
epoch: 1036 train_loss 0.010094799780442068 valid_loss 0.015496572169164816
epoch: 1037 train_loss 0.010097014946707835 valid_loss 0.015496351872570813
epoch: 1038 train_loss 0.01009923431944723 valid_loss 0.015496128735442956
epoch: 1039 train_loss 0.010101452597882598 valid_loss 0.015495890953267614
epoch: 1040 train_loss 0.010103671186758827 valid_loss 0.015495663365193953
epoch: 1041 train_loss 0.010105883974271515 valid_loss 0.015495443545902769
epoch: 1042 train_loss 0.01010810083632047 valid_loss 0.015495218910897772
epoch: 1043 train_loss 0.01011031597154215 valid_loss 0.015494995176171263
epoch: 1044 train_loss 0.010112527722958476 valid_loss 0.015494773575725655
epoch: 1045 train_loss 0.010114740599722912 valid_loss 0.015494567039422692
epoch: 1046 train_loss 0.010116950558343281 valid_loss 0.015494342202631135
epoch: 1047 train_loss 0.010119164362549782 valid_loss 0.015494136066020776
epoch: 1048 train_loss 0.010121371503919363 valid_loss 0.015493926654259364
epoch: 1049 train_loss 0.010123586274373035 valid_loss 0.015493718433814745
epoch: 1050 train_loss 0.01012579408319046 valid_loss 0.01549351349240169
epoch: 1051 train_loss 0.010128000704571604 valid_loss 0.01549331514009585
epoch: 1052 train_loss 0.010130208226231237 valid_loss 0.01549311246102055
epoch: 1053 train_loss 0.010132416271759818 valid_loss 0.015492915571667255
epoch: 1054 train_loss 0.010134621371980757 valid_loss 0.015492712294993301
epoch: 1055 train_loss 0.010136826216087988 valid_loss 0.01549252774566412
epoch: 1056 train_loss 0.010139032340763758 valid_loss 0.015492322544256846
epoch: 1057 train_loss 0.010141236781297873 valid_loss 0.015492133260704578
epoch: 1058 train_loss 0.010143437838026632 valid_loss 0.015491949363301198
epoch: 1059 train_loss 0.010145640287858745 valid_loss 0.01549176179493467
epoch: 1060 train_loss 0.010147842726049324 valid_loss 0.015491573951051881
epoch: 1061 train_loss 0.010150042195649196 valid_loss 0.015491387139384945
epoch: 1062 train_loss 0.010152240567064534 valid_loss 0.01549120523268357
epoch: 1063 train_loss 0.010154439182952047 valid_loss 0.01549103199892367
epoch: 1064 train_loss 0.01015663775227343 valid_loss 0.015490855521056801
epoch: 1065 train_loss 0.010158835638624927 valid_loss 0.015490686881821602
epoch: 1066 train_loss 0.010161034929721306 valid_loss 0.015490499511361121
epoch: 1067 train_loss 0.010163231822662055 valid_loss 0.015490337597051014
epoch: 1068 train_loss 0.010165425029117613 valid_loss 0.015490163730767866
epoch: 1069 train_loss 0.010167617820358524 valid_loss 0.01548999579778562
epoch: 1070 train_loss 0.01016981234618773 valid_loss 0.015489835563736657
epoch: 1071 train_loss 0.01017200833496948 valid_loss 0.015489668434020132
epoch: 1072 train_loss 0.010174198805664976 valid_loss 0.015489506546873599
epoch: 1073 train_loss 0.010176391853019596 valid_loss 0.015489349913938593
epoch: 1074 train_loss 0.010178583134741832 valid_loss 0.01548919314906622
```

```
epoch: 1075 train_loss 0.010180775976429383 valid_loss 0.015489033772610128
epoch: 1076 train_loss 0.01018296245407934 valid_loss 0.015488889099409183
epoch: 1077 train_loss 0.010185149820366253 valid_loss 0.015488733906143656
epoch: 1078 train_loss 0.010187340438521156 valid_loss 0.01548858309785525
epoch: 1079 train_loss 0.010189525973207008 valid_loss 0.015488438754497716
epoch: 1080 train_loss 0.010191711701918394 valid_loss 0.0154882875077116
epoch: 1081 train_loss 0.010193897360780586 valid_loss 0.015488148434087634
epoch: 1082 train_loss 0.010196083046806356 valid_loss 0.015488000326634695
epoch: 1083 train_loss 0.010198271158151329 valid_loss 0.01548786675169443
epoch: 1084 train_loss 0.010200453751410047 valid_loss 0.015487723106828829
epoch: 1085 train_loss 0.01020263652317226 valid_loss 0.015487582853529603
epoch: 1086 train_loss 0.010204816015902906 valid_loss 0.015487460213868568
epoch: 1087 train_loss 0.010206995756986241 valid_loss 0.015487321671874573
epoch: 1088 train_loss 0.010209174923753986 valid_loss 0.01548718656413257
epoch: 1089 train_loss 0.010211357920585822 valid_loss 0.01548705951621135
epoch: 1090 train_loss 0.010213538344639042 valid_loss 0.015486934750030438
epoch: 1091 train_loss 0.0102157178412502 valid_loss 0.0154868099023588
epoch: 1092 train_loss 0.010217894423597803 valid_loss 0.015486684115603565
epoch: 1093 train_loss 0.01022006853017956 valid_loss 0.015486564657961329
epoch: 1094 train_loss 0.01022224397941803 valid_loss 0.01548644684177513
epoch: 1095 train_loss 0.01022441991372034 valid_loss 0.015486326375200104
epoch: 1096 train_loss 0.010226595533701281 valid_loss 0.015486207767389714
epoch: 1097 train_loss 0.01022876868955791 valid_loss 0.01548609189145888
epoch: 1098 train_loss 0.010230943572241812 valid_loss 0.015485989364484946
epoch: 1099 train_loss 0.010233114652025203 valid_loss 0.01548587557238837
epoch: 1100 train_loss 0.010235285440770289 valid_loss 0.015485770895611494
epoch: 1101 train_loss 0.01023745610145852 valid_loss 0.015485657774843276
epoch: 1102 train_loss 0.010239624779205769 valid_loss 0.015485553920734673
epoch: 1103 train_loss 0.010241795482579618 valid_loss 0.015485444529137265
epoch: 1104 train_loss 0.010243963073783864 valid_loss 0.015485353589368363
epoch: 1105 train_loss 0.010246134382517388 valid_loss 0.015485245420131832
epoch: 1106 train_loss 0.010248295962810516 valid_loss 0.015485148287067811
epoch: 1107 train_loss 0.010250463678191105 valid_loss 0.015485049252553533
epoch: 1108 train_loss 0.010252633671431493 valid_loss 0.015484965441282838
epoch: 1109 train_loss 0.010254795212919514 valid_loss 0.015484857745468617
epoch: 1110 train_loss 0.010256961760266374 valid_loss 0.015484777244273574
epoch: 1111 train_loss 0.010259125098430862 valid_loss 0.015484697763652851
epoch: 1112 train_loss 0.0102612836790892 valid_loss 0.015484600638349852
epoch: 1113 train_loss 0.010263447316053014 valid_loss 0.01548451230628416
epoch: 1114 train_loss 0.010265607056984058 valid_loss 0.015484429767820985
epoch: 1115 train_loss 0.010267765164220085 valid_loss 0.015484351940297831
epoch: 1116 train_loss 0.010269929522958894 valid_loss 0.01548427134209002
epoch: 1117 train_loss 0.010272088817631204 valid_loss 0.01548418893556421
```

```
epoch: 1118 train_loss 0.010274246560099225 valid_loss 0.015484109687774132
epoch: 1119 train_loss 0.01027640311513096 valid_loss 0.015484039850222567
epoch: 1120 train_loss 0.010278560768347234 valid_loss 0.015483963621469836
epoch: 1121 train_loss 0.010280715945797662 valid_loss 0.01548390401682506
epoch: 1122 train_loss 0.010282871945916365 valid_loss 0.015483814291656017
epoch: 1123 train_loss 0.010285027899468939 valid_loss 0.015483758594685544
epoch: 1124 train_loss 0.010287177411373704 valid_loss 0.015483682552197326
epoch: 1125 train_loss 0.010289330667971323 valid_loss 0.01548362480631719
epoch: 1126 train_loss 0.010291483610247572 valid_loss 0.015483554716532429
epoch: 1127 train_loss 0.010293634848979612 valid_loss 0.015483500284608454
epoch: 1128 train_loss 0.010295785265043378 valid_loss 0.015483434622486433
epoch: 1129 train_loss 0.01029793821896116 valid_loss 0.015483378696565827
epoch: 1130 train_loss 0.010300084463475892 valid_loss 0.015483326503696541
epoch: 1131 train_loss 0.010302237906338026 valid_loss 0.015483276379139472
epoch: 1132 train_loss 0.010304384597111494 valid_loss 0.015483207344853629
epoch: 1133 train_loss 0.01030652632083123 valid_loss 0.015483165672048927
epoch: 1134 train_loss 0.01030867213072876 valid_loss 0.015483119032190492
epoch: 1135 train_loss 0.01031082309394454 valid_loss 0.015483063273131847
epoch: 1136 train_loss 0.01031296478273968 valid_loss 0.01548302184867983
epoch: 1137 train_loss 0.010315110410253207 valid_loss 0.01548296941909939
epoch: 1138 train_loss 0.010317251652789612 valid_loss 0.015482932464995732
epoch: 1139 train_loss 0.010319395662130167 valid_loss 0.015482889395207167
epoch: 1140 train_loss 0.010321537715693314 valid_loss 0.015482846666903546
epoch: 1141 train_loss 0.010323674592655152 valid_loss 0.01548281495925039
epoch: 1142 train_loss 0.010325817887981733 valid_loss 0.01548276887430499
epoch: 1143 train_loss 0.010327958855001877 valid_loss 0.015482736487562458
epoch: 1144 train_loss 0.010330093970211843 valid_loss 0.015482701283569137
epoch: 1145 train_loss 0.010332229888687532 valid_loss 0.015482668415643275
epoch: 1146 train_loss 0.010334371755986164 valid_loss 0.015482645493466406
epoch: 1147 train_loss 0.010336508462205529 valid_loss 0.01548261259837697
epoch: 1148 train_loss 0.010338643666667242 valid_loss 0.015482580071936052
epoch: 1149 train_loss 0.010340781932851921 valid_loss 0.01548255286179483
epoch: 1150 train_loss 0.01034290959360078 valid_loss 0.015482536160076658
epoch: 1151 train_loss 0.010345047192337612 valid_loss 0.015482507890556007
epoch: 1152 train_loss 0.01034717815928161 valid_loss 0.01548249153032278
epoch: 1153 train_loss 0.010349311931834866 valid_loss 0.015482472282989572
epoch: 1154 train_loss 0.010351443383842706 valid_loss 0.015482446586247534
epoch: 1155 train_loss 0.010353577222364645 valid_loss 0.015482436787957946
epoch: 1156 train_loss 0.010355703874180715 valid_loss 0.015482422472753872
epoch: 1157 train_loss 0.010357833583839238 valid_loss 0.015482407171900074
epoch: 1158 train_loss 0.010359963685429345 valid_loss 0.015482397482264787
epoch: 1159 train_loss 0.010362092358991503 valid_loss 0.015482388436794281
epoch: 1160 train_loss 0.010364215107013781 valid_loss 0.015482375200372189
```

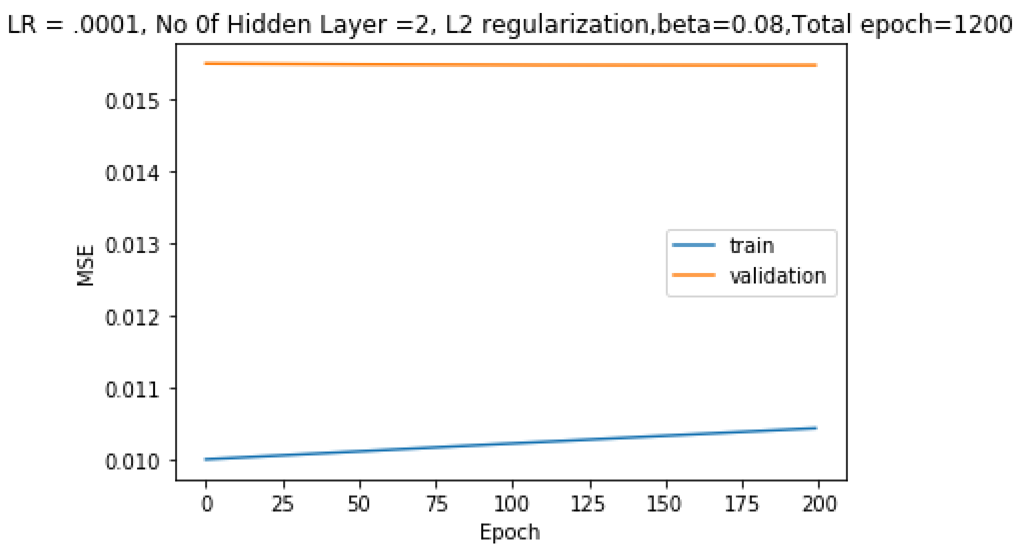
```
epoch: 1161 train_loss 0.010366345068905502 valid_loss 0.015482371653585384
epoch: 1162 train_loss 0.010368466291887065 valid_loss 0.01548236411763355
epoch: 1163 train_loss 0.010370593921591838 valid_loss 0.015482364521206667
epoch: 1164 train_loss 0.01037271690632527 valid_loss 0.015482355685283741
epoch: 1165 train_loss 0.01037484254144753 valid_loss 0.015482360923973223
epoch: 1166 train_loss 0.010376962445055446 valid_loss 0.015482364563892285
epoch: 1167 train_loss 0.010379086434841155 valid_loss 0.015482359880115837
epoch: 1168 train_loss 0.010381207545287907 valid_loss 0.015482364548370242
epoch: 1169 train_loss 0.010383325989823788 valid_loss 0.015482368809171022
epoch: 1170 train_loss 0.010385444251975665 valid_loss 0.01548237974445025
epoch: 1171 train_loss 0.010387564466024438 valid_loss 0.01548238667504241
epoch: 1172 train_loss 0.010389685483338932 valid_loss 0.015482397315402826
epoch: 1173 train_loss 0.010391802232091626 valid_loss 0.015482406628628572
epoch: 1174 train_loss 0.010393915612561007 valid_loss 0.015482415336494644
epoch: 1175 train_loss 0.010396036874347677 valid_loss 0.015482433939663072
epoch: 1176 train_loss 0.010398145652531336 valid_loss 0.015482444999118645
epoch: 1177 train_loss 0.010400265800611427 valid_loss 0.01548246395541355
epoch: 1178 train_loss 0.01040237924316898 valid_loss 0.015482486070444187
epoch: 1179 train_loss 0.01040449074159066 valid_loss 0.01548249441742276
epoch: 1180 train_loss 0.010406605895453443 valid_loss 0.01548252688953653
epoch: 1181 train_loss 0.010408712349211176 valid_loss 0.015482546016573905
epoch: 1182 train_loss 0.01041082749919345 valid_loss 0.015482573956251144
epoch: 1183 train_loss 0.010412939152835558 valid_loss 0.01548260086371253
epoch: 1184 train_loss 0.010415045222422729 valid_loss 0.015482629300095141
epoch: 1185 train_loss 0.010417155413112293 valid_loss 0.015482657856773585
epoch: 1186 train_loss 0.010419261172258606 valid_loss 0.015482682773532966
epoch: 1187 train_loss 0.010421368448684612 valid_loss 0.01548271613040318
epoch: 1188 train_loss 0.010423479966508846 valid_loss 0.015482752723619342
epoch: 1189 train_loss 0.010425581166055053 valid_loss 0.015482778816173473
epoch: 1190 train_loss 0.010427687010572602 valid_loss 0.0154828135157004
epoch: 1191 train_loss 0.010429791531835993 valid_loss 0.015482863073702901
epoch: 1192 train_loss 0.010431895094613234 valid_loss 0.01548289906932041
epoch: 1193 train_loss 0.010433997543683897 valid_loss 0.01548293575566883
epoch: 1194 train_loss 0.010436100229465712 valid_loss 0.015482985170092434
epoch: 1195 train_loss 0.010438199667260051 valid_loss 0.015483018363981197
epoch: 1196 train_loss 0.010440296435263008 valid_loss 0.01548305842637395
epoch: 1197 train_loss 0.010442401503678412 valid_loss 0.015483115865693738
epoch: 1198 train_loss 0.010444498838235934 valid_loss 0.015483156226885815
epoch: 1199 train_loss 0.010446599933008354 valid_loss 0.015483202323472748
```

In [42]:
```
print(np.sqrt(loss_tr[-1]),np.sqrt(loss_va[-1]))
```

```
0.10220860987709575 0.12443151660038845
```

Plotting MSE for train and validation

In [46]:
```
#Plotting
plt.title('LR = .0001, No 0f Hidden Layer =2, L2 regularization,beta=0.08,Total epoch=1200')
plt.plot(loss_tr[1000:], label = 'train')
plt.plot(loss_va[1000:],label='validation')
plt.xlabel('Epoch')
plt.ylabel('MSE')
plt.legend()
plt.show()
```



4.4. Result of best model:

In [47]:
```
#Prediction on test data
in_test={input_feat:np.array(test_df)}
pred = sess.run((out),feed_dict=in_test)
pred = pred.reshape(-1)
```

In [49]:
```
#Wrting Submission file
pred_exp=np.exp(pred)
print(len(pred_exp))
data='Id,SalePrice\n'
Id=1461
for row in pred_exp:
    data+=str(Id)+','+str(row)+'\n'
    Id+=1
file=open('HP_3layerNN.csv','w')
file.write(data)
file.close()
```

```
1459
```

NOTE:

Models were tuned using 70% training data for training and 30% training data for validation. Kaggle scores are based on 99% training data.

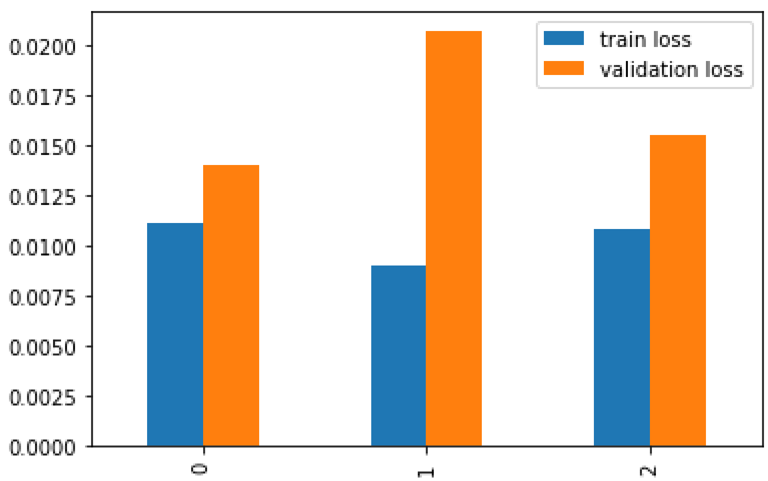
Cross validation

Cross validation for the best model for 3 fold has been shown below:

In [1]:
```
LearningRate = .0001
epoch = 1200
reg_const = .08
from sklearn.model_selection import KFold
k_fold=KFold(n_splits=3)
splited_index = k_fold.split(train_df)
i =0
loss_df = pd.DataFrame()
for tr_ind,ts_ind in splited_index:
    train_data=np.array_split(train_df[tr_ind],no_of_batch)
    train_lb=np.array_split(labels_train[tr_ind],no_of_batch)
    val_data=np.array_split(train_df[ts_ind],no_of_batch)
    val_lb=np.array_split(labels_train[ts_ind],no_of_batch)
    input_feat,sess,out,loss_tr_cv,loss_va_cv=train_loop(LearningRate,epoch,reg_const,train_data,train_lb,val
_data,val_lb)
    tr_loss = np.mean(loss_tr_cv[-10:])
    valid_loss = np.mean(loss_va_cv[-10:])
    loss_df.loc[i,'train loss']= tr_loss
    loss_df.loc[i,'validation loss']= valid_loss
    i = i+1
print(loss_df)
```

In [51]:
```
loss_df.plot(kind = 'bar')
```

Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x1faff3f37b8>



5. Decision,Result and discussion:

5.1. Decision:

Best result was found for 2 hidden layer model (Kaggle score=.12140).So final model is a 3 layer NN model.Among all the experiments,best three ANN models are presented below with screenshots from Kaggle.

5.2. Kaggle result for ANN models

5.2.1. No hidden layer:

HousePriceANNmodel.csv 0.12381

4 minutes ago by Sumaiya Saima

Model: ANN with no hidden layer, L2 Reg. Const = 0.1, Learning rate = 0.001, Epoch = 250

5.2.2. One hidden layer:

HousePriceHL1tuned.csv 0.12605

an hour ago by Sumaiya Saima

Hidden Layer: 1, LR: 0.0001, RC = 0.1, No of Batch = 80

5.2.3. Two hidden layer: (Model with best Kaggle result)

HP_3layerNN.csv 0.12140
4 minutes ago by Sumaiya Saima
Hidden Layer: 2, Epoch: 1200, LR = .0001, RC = .08, No of batch = 60

Note: DNN models with more hidden layers showed in the hyper parameter tuning part could not achieve better result than these three models in Kaggle.

5.3 Discussion:

Why models with no activation function performed well?

- A important point to note here is that a linear regression model(tuned with Grid Search of Scikit Learns library) was used to solve this problem which performed a lot better(kaggle score=.11814) than the NN model.

HousePriceLR.csv 0.11814
4 minutes ago by Sumaiya Saima
Ridge(alpha=10), CV = 10

- Since an NN model with no activation function actually behaves like a linear regression model, so it performs better than other NN models as expected. Incorporating activation functions forces non-linearity on the model which is not always required specially in this case.

Why additional hidden layers are not needed?

- Adding more layers did not improve the result. One issue within this subject on which there is a consensus is the performance difference from adding additional hidden layers: the situations in which performance improves with a second (or third, etc.) hidden layer are very few.[6] Moreover it unnecessarily makes a model more complex. So as a combined effect, adding more hidden layers with higher values of epochs and various ranges of other hyperparameters(learning rate, regularization constants etc) did not really result in any improvements.

Why regularization is needed?

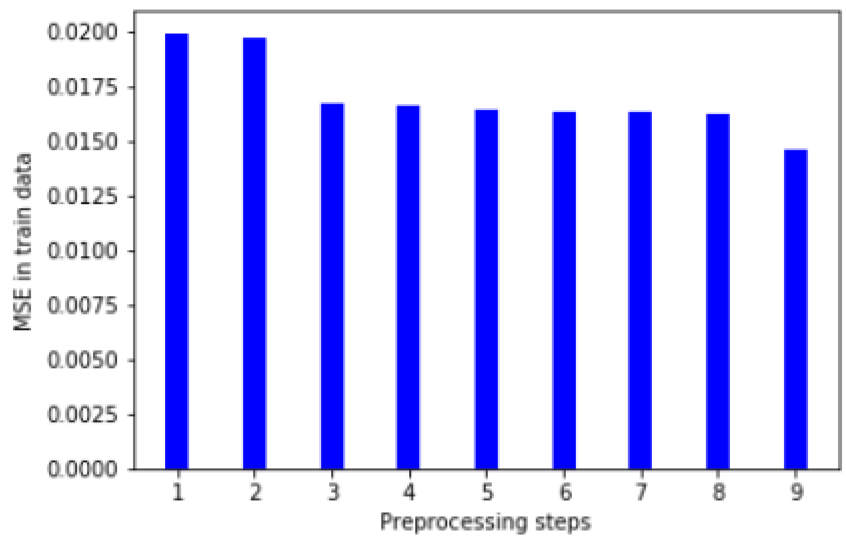
- Regularization ensures the model is not overfiited. The graphs showed above also support this statement. So to ensure a satisfactory model, any ANN should be used with regulaization.
- Choice of L1 or L2 regularization depends on the fact if all the features are must to train a model. If we want to find best features for a model and only use those to train the model, L1 regularization is recommended. Otherwise L2 regularization should be used as it never sets any weight values to zero and thus never fully elliminate any feature.

Why is preprocessing needed?

- Adding various stages of preprocessing gradually improves the model. A barplot has been shown to visualize the effects of preprocessing.

```
1 :  Only Target unskewed
2 :  With RobustScale
3 :  OutlierHandled
4 :  DroppedGarARea
5 :  Dropped1stFlr
6 :  DroppedTotRmsAbv
7 :  MissingHandled
8 :  NewFeat
9 :  AllFeatSkwed

Effect of preprocessing in MSE
```



Conclusion:

- In this kind of problem, it is really important to understand the data and preprocess it accordingly.More accurate and logical preprocessing would result in a model with better accuracy.
- The performance of the models presented here are stable.Among them the best model was chosen with detail observations and the result and behaviour have been explained properly. so the NN model presented here can be claimed a simple enough model with satisfactory results.

References:

1. http://fmwww.bc.edu/repec/bocode/t/transint.html (http://fmwww.bc.edu/repec/bocode/t/transint.html)
2. https://en.wikipedia.org/wiki/Outlier (https://en.wikipedia.org/wiki/Outlier)
3. https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba (https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba)
4. https://developerzen.com/data-mining-handling-missing-values-the-database-bd2241882e72 (https://developerzen.com/data-mining-handling-missing-values-the-database-bd2241882e72)
5. https://medium.com/diogo-menezes-borges/project-2-predicting-house-prices-on-kaggle-989f1b0c4ef6 (https://medium.com/diogo-menezes-borges/project-2-predicting-house-prices-on-kaggle-989f1b0c4ef6)
6. https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw/1097#1097 (https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw/1097#1097)

Note:

Please use the content of this notebook for learning purposes only.

Acknowledgement:

Thanks and gratitude to Md.Saiful Islam(Lecturer,DEPT of CSE,BUET) and Chowdhury Md. Rakin Haider(Lecturer,DEPT of CSE,BUET) for their guidance throughout the assignment.

In [ ]:

```

```