In [1]:
```
from fastai.tabular import *
```

In [2]:
```
pwd
```

Out[2]:
```
'D:\\python\\projects'
```

In [3]:
```
df = pd.read_csv('train.csv')
```

In [4]:
```
df.shape
```

Out[4]:
```
(1460, 80)
```

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

Out[5]:
```
Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyl
e',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBl
t',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCon
d',
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorc
h',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVa
l',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [6]:
```
df['SalePrice']
```

Out[6]:
```
0       208500.0
1       181500.0
2       223500.0
3       140000.0
4       250000.0
          ...
1455    175000.0
1456    210000.0
1457    266500.0
1458    142125.0
1459    147500.0
Name: SalePrice, Length: 1460, dtype: float64
```

In [7]:
```
cat_names = ['MSSubClass', 'MSZoning','Street','Alley','LotShape','LandContou
r','Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType','Ho
useStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType'
,'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'BsmtFinSF
2', 'Heating','Fireplaces', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQ
ual',
       'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual'
,'GarageCond', 'PavedDrive',  'PoolQC', 'Fence', 'MiscFeature', 'SaleType',
       'SaleCondition']

# cont_names = ['LotFrontage', 'SalePrice']
cont_names = ['LotFrontage', 'LotArea', 'OverallQual', 'OverallCond', 'YearBui
lt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtUnfSF', 'TotalBsmtSF',
'1stFlrSF', '2ndFlrSF',        'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'B
smtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsA
bvGrd',        'GarageYrBlt','GarageCars', 'GarageArea','WoodDeckSF', 'OpenPor
chSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea','MiscVal', 'MoS
old', 'YrSold']
```

In [8]:
```
procs = [FillMissing, Categorify, Normalize]
```

In [9]:
```
dep_var = 'SalePrice'
```

In [10]:
```
TabularList.from_df?
```

In [11]:
```
pwd
```

Out[11]:
```
'D:\\python\\projects'
```

In [31]:
```
path="D:\\python\\projects"
```

In [53]:
```
testDf = pd.read_csv('test.csv')
testDf = testDf.drop('Id',axis=1)
```

In [54]:
```
test = TabularList.from_df(testDf, path=path, cat_names=cat_names, cont_names=
cont_names, procs=procs)
```

In [55]:
```
data = (TabularList.from_df(df, path=path, cat_names=cat_names, cont_names=con
t_names, procs=procs)
                           .split_by_rand_pct(valid_pct = 0.2)
                           .label_from_df(cols=dep_var)
                           .add_test(test)
                           .databunch())
```

In [56]:
```
data.show_batch(rows=10)
```

MSSubClass	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
20	RL	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl
60	RL	Pave	#na#	IR1	Lvl	AllPub	Inside	Gtl
20	RL	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl
60	RL	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl
190	RL	Pave	#na#	Reg	Lvl	AllPub	Corner	Gtl
120	RL	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl
60	RL	Pave	#na#	IR2	Lvl	AllPub	CulDSac	Gtl
190	RL	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl
20	RL	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl
50	RM	Pave	#na#	Reg	Lvl	AllPub	Inside	Gtl

In [57]:
```
learn = tabular_learner(data, layers=[200,100], metrics=mean_squared_error)
```

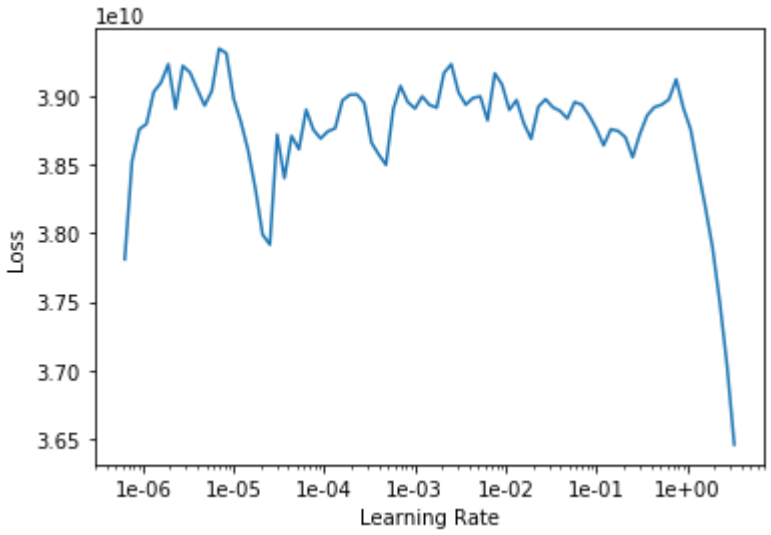
In [58]:
```
learn.fit(1, 1e-2)
```

epoch	train_loss	valid_loss	mean_squared_error	time
0	38454759424.000000	39784558592.000000	39784558592.000000	00:33

In [59]:
```
learn.lr_find()
```

```
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```

In [60]:
```
learn.recorder.plot()
```



In [61]:
```
learn.unfreeze()
```

In [62]:
```
learn.fit_one_cycle(8, max_lr=slice(1))
```

epoch	train_loss	valid_loss	mean_squared_error	time
0	38928265216.000000	40127021056.000000	40127021056.000000	00:32
1	36535541760.000000	30608543744.000000	30608543744.000000	00:32
2	25556934656.000000	11353831424.000000	11353831424.000000	00:31
3	16017620992.000000	697410304.000000	697410304.000000	00:33
4	10600801280.000000	1344547712.000000	1344547712.000000	00:32
5	7287837184.000000	1656420992.000000	1656420992.000000	00:32
6	5127537664.000000	1007617792.000000	1007617792.000000	00:32
7	3729403136.000000	792713984.000000	792713984.000000	00:32

In [63]:
```
preds, _ = learn.get_preds(ds_type=DatasetType.Test)
```

In [64]:
```
preds
```

Out[64]:
```
tensor([[116488.9375],
        [148379.1875],
        [163991.6719],
        ...,
        [149667.2812],
        [106791.6562],
        [196611.8750]])
```

In [65]:
```
result = preds.numpy()[:, 0]
result
```

Out[65]:
```
array([116488.94 , 148379.19 , 163991.67 , 179681.6  , ...,  74694.664, 14966
7.28 , 106791.66 , 196611.88 ],
      dtype=float32)
```

In [66]:
```
ids = [i for i in range(1461,2920)]
```

In [67]:
```
d = {'Id': ids, 'SalePrice':result}
df2 = DataFrame(data=d)
```

In [68]:
```
df2
```

Out[68]:

	Id	SalePrice
0	1461	116488.937500
1	1462	148379.187500
2	1463	163991.671875
3	1464	179681.593750
4	1465	181262.062500
5	1466	152568.203125
6	1467	162908.093750
7	1468	149027.234375
8	1469	169962.500000
9	1470	119869.390625
10	1471	169823.062500
11	1472	94735.507812
12	1473	92395.976562
13	1474	131670.765625
14	1475	110182.046875
15	1476	337405.125000
16	1477	246726.953125
17	1478	237359.484375
18	1479	294626.875000
19	1480	458572.531250
20	1481	307000.687500
21	1482	199689.843750
22	1483	156071.765625
23	1484	157050.015625
24	1485	169191.921875
25	1486	178222.000000
26	1487	317039.031250
27	1488	213186.609375
28	1489	186340.312500
29	1490	211604.890625
...	...	...
1429	2890	80093.531250
1430	2891	127738.023438
1431	2892	20538.244141
1432	2893	65455.601562

	Id	SalePrice
1433	2894	48938.390625
1434	2895	342943.906250
1435	2896	287359.000000
1436	2897	190548.093750
1437	2898	139768.906250
1438	2899	188175.203125
1439	2900	148728.781250
1440	2901	163470.796875
1441	2902	189761.234375
1442	2903	316707.375000
1443	2904	344295.781250
1444	2905	83627.109375
1445	2906	179758.500000
1446	2907	102104.328125
1447	2908	127977.953125
1448	2909	139759.781250
1449	2910	76957.484375
1450	2911	79710.875000
1451	2912	139570.703125
1452	2913	75097.828125
1453	2914	73883.210938
1454	2915	79570.304688
1455	2916	74694.664062
1456	2917	149667.281250
1457	2918	106791.656250
1458	2919	196611.875000

1459 rows × 2 columns

In [69]:
```
df2.to_csv('out.csv', index=False)
```

In [ ]:
```
```