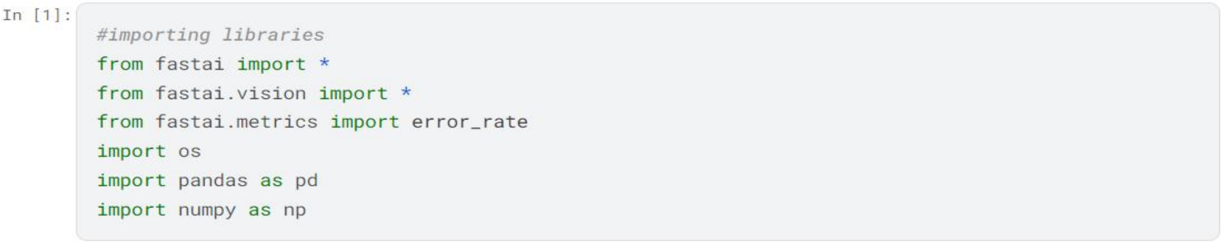
PNEUMONIA DETECTION USING CNN

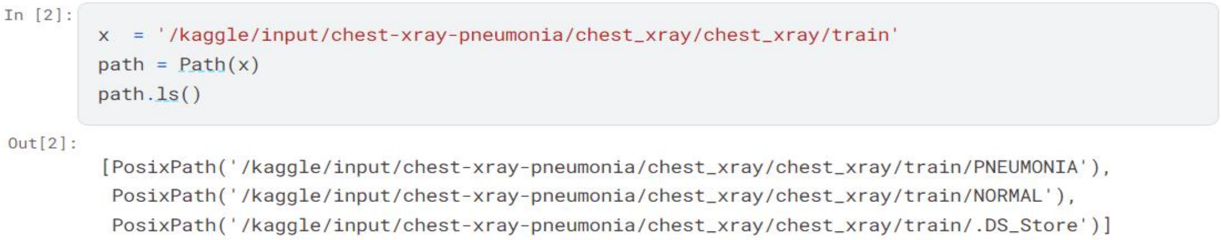
Objective :

The aim of this kernel is to provide all the tips and tricks required to train image classification model on Pneumonia image dataset in a single page.This kernel will hold almost all steps and steps required to implement image classification algorithm using SOTA such as ResNET on Pnemonia Dataset.It could be a great time saver for you.Just utilize it anytime when you are working on Image Classification.



```
In [1]:
#importing libraries
from fastai import *
from fastai.vision import *
from fastai.metrics import error_rate
import os
import pandas as pd
import numpy as np
```

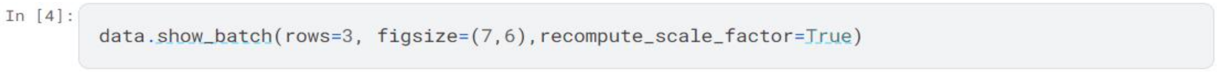
Setting up path for training data :



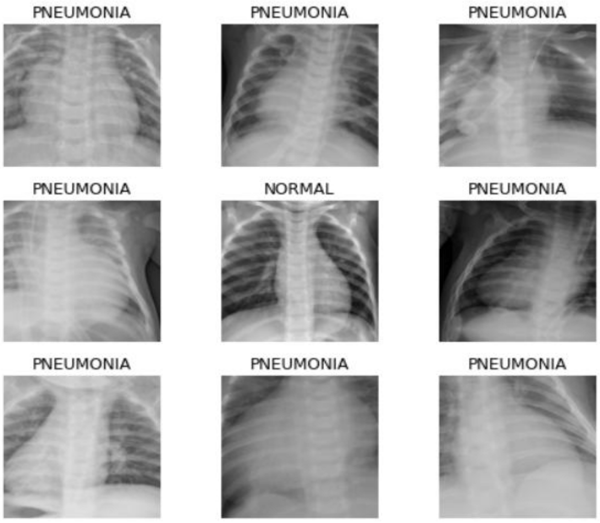
```
In [2]:
x  = '/kaggle/input/chest-xray-pneumonia/chest_xray/chest_xray/train'
path = Path(x)
path.ls()
```

```
Out[2]:
[PosixPath('/kaggle/input/chest-xray-pneumonia/chest_xray/chest_xray/train/PNEUMONIA'),
 PosixPath('/kaggle/input/chest-xray-pneumonia/chest_xray/chest_xray/train/NORMAL'),
 PosixPath('/kaggle/input/chest-xray-pneumonia/chest_xray/chest_xray/train/.DS_Store')]
```

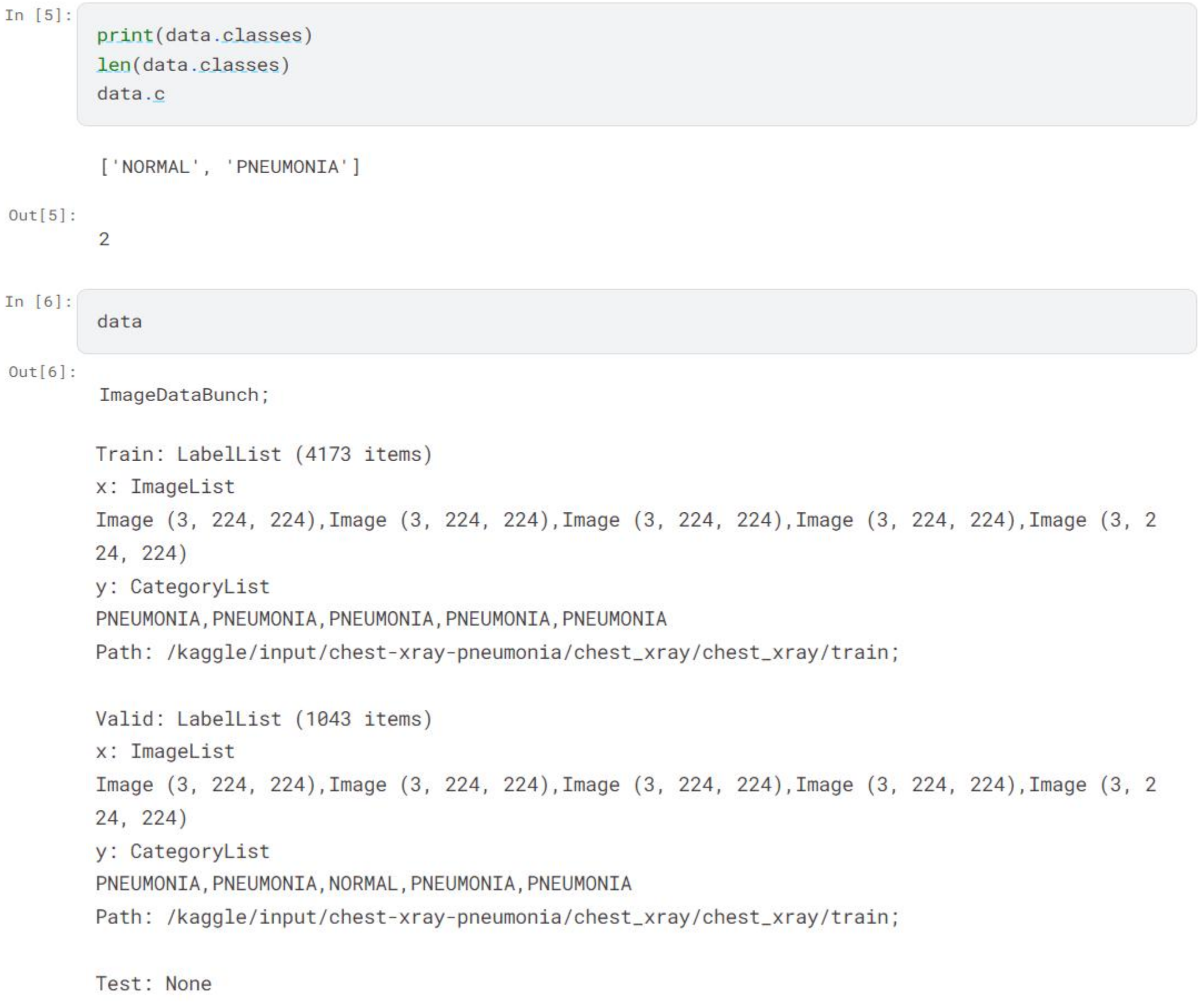
Data Explorations :



```
In [4]:
data.show_batch(rows=3, figsize=(7,6),recompute_scale_factor=True)
```



Page 2



In [5]:

```
print(data.classes)
len(data.classes)
data.c
```

```
['NORMAL', 'PNEUMONIA']
```

Out[5]:

```
2
```

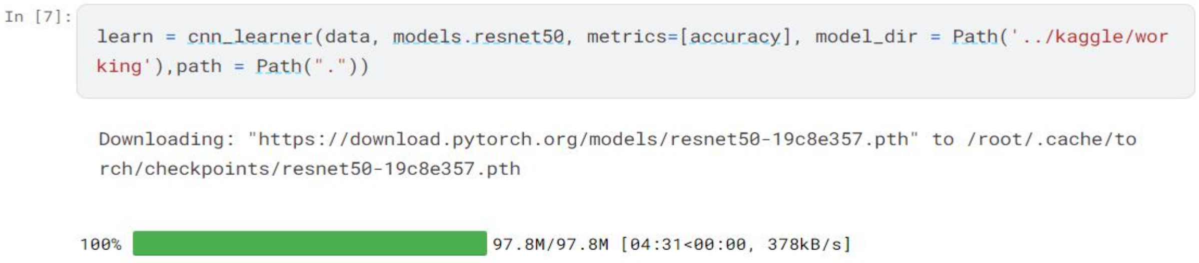
In [6]:

```
data
```

Out[6]:

```
ImageDataBunch;

Train: LabelList (4173 items)
x: ImageList
Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image (3, 2
24, 224)
y: CategoryList
PNEUMONIA,PNEUMONIA,PNEUMONIA,PNEUMONIA,PNEUMONIA
Path: /kaggle/input/chest-xray-pneumonia/chest_xray/chest_xray/train;

Valid: LabelList (1043 items)
x: ImageList
Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image (3, 2
24, 224)
y: CategoryList
PNEUMONIA,PNEUMONIA,NORMAL,PNEUMONIA,PNEUMONIA
Path: /kaggle/input/chest-xray-pneumonia/chest_xray/chest_xray/train;

Test: None
```

Creating Model :



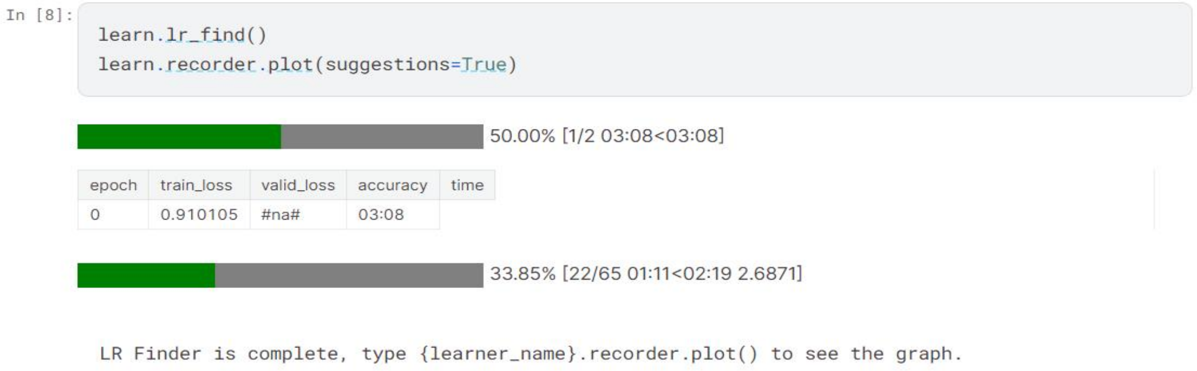
In [7]:

```
learn = cnn_learner(data, models.resnet50, metrics=[accuracy], model_dir = Path('../kaggle/wor
king'),path = Path("."))
```

```
Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /root/.cache/to
rch/checkpoints/resnet50-19c8e357.pth
```

```
100% 97.8M/97.8M [04:31<00:00, 378kB/s]
```

Finding LR :



In [8]:

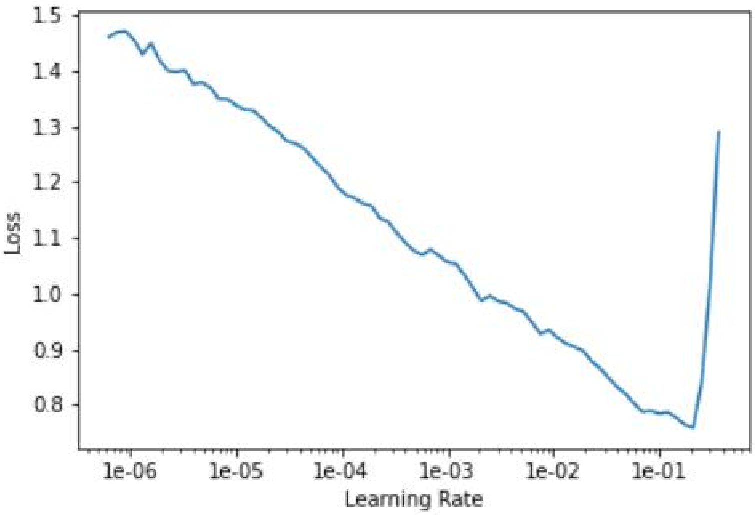
```
learn.lr_find()
learn.recorder.plot(suggestions=True)
```

```
50.00% [1/2 03:08<03:08]
```

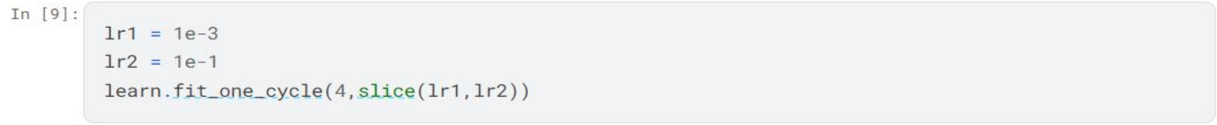
epoch	train_loss	valid_loss	accuracy	time
0	0.910105	#na#	03:08	

```
33.85% [22/65 01:11<02:19 2.6871]
```

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

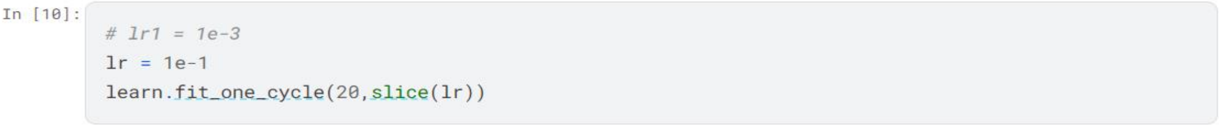


Train Model :

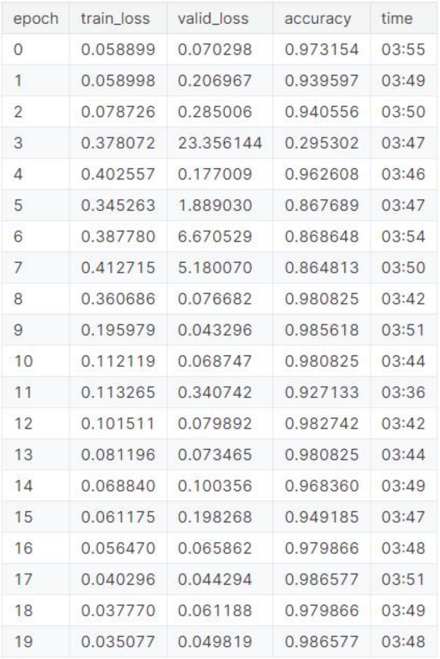


```
In [9]:
lr1 = 1e-3
lr2 = 1e-1
learn.fit_one_cycle(4,slice(lr1,lr2))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.531993	0.601826	0.920422	03:54
1	0.351373	0.306143	0.932886	03:49
2	0.198085	0.039864	0.985618	03:49
3	0.099993	0.071525	0.975072	03:48

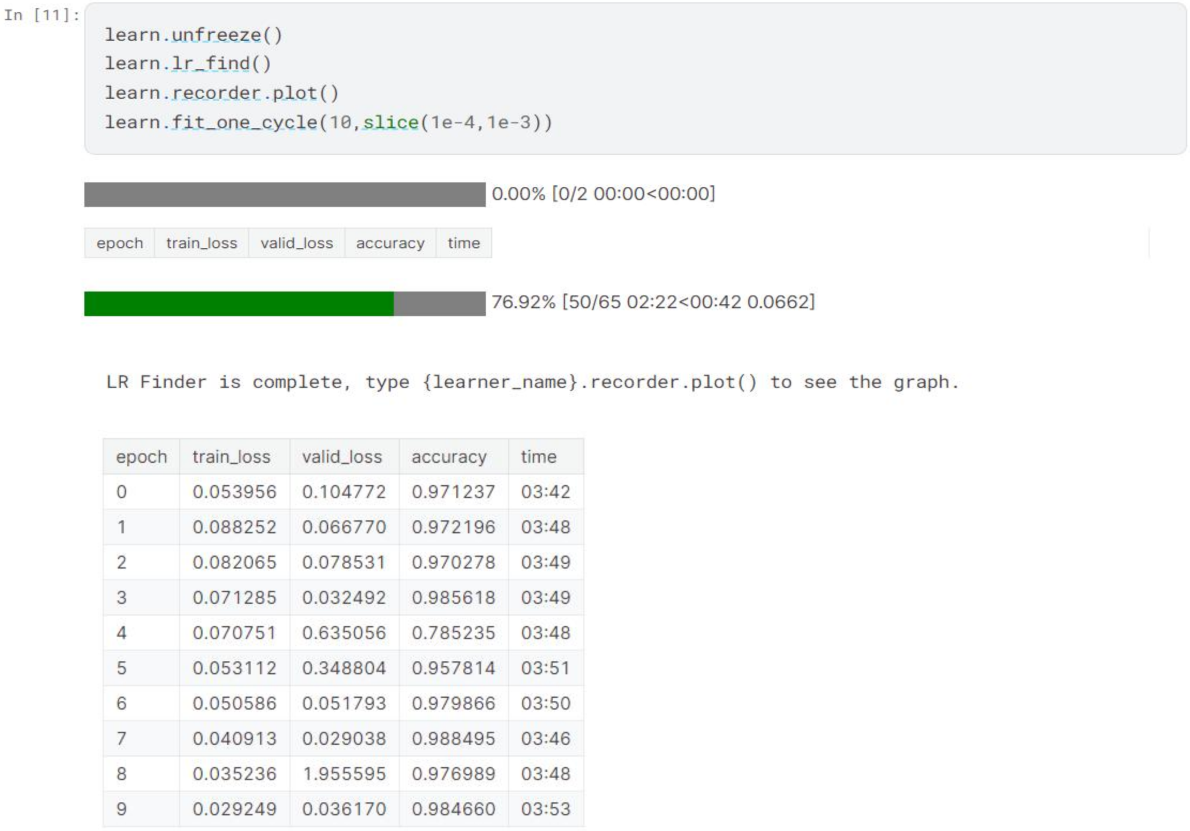


```
In [10]:
# lr1 = 1e-3
lr = 1e-1
learn.fit_one_cycle(20,slice(lr))
```



epoch	train_loss	valid_loss	accuracy	time
0	0.058899	0.070298	0.973154	03:55
1	0.058998	0.206967	0.939597	03:49
2	0.078726	0.285006	0.940556	03:50
3	0.378072	23.356144	0.295302	03:47
4	0.402557	0.177009	0.962608	03:46
5	0.345263	1.889030	0.867689	03:47
6	0.387780	6.670529	0.868648	03:54
7	0.412715	5.180070	0.864813	03:50
8	0.360686	0.076682	0.980825	03:42
9	0.195979	0.043296	0.985618	03:51
10	0.112119	0.068747	0.980825	03:44
11	0.113265	0.340742	0.927133	03:36
12	0.101511	0.079892	0.982742	03:42
13	0.081196	0.073465	0.980825	03:44
14	0.068840	0.100356	0.968360	03:49
15	0.061175	0.198268	0.949185	03:47
16	0.056470	0.065862	0.979866	03:48
17	0.040296	0.044294	0.986577	03:51
18	0.037770	0.061188	0.979866	03:49
19	0.035077	0.049819	0.986577	03:48

Hyper Parameter Tuning :

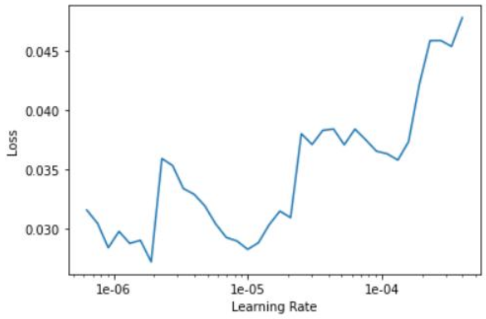


In [11]:

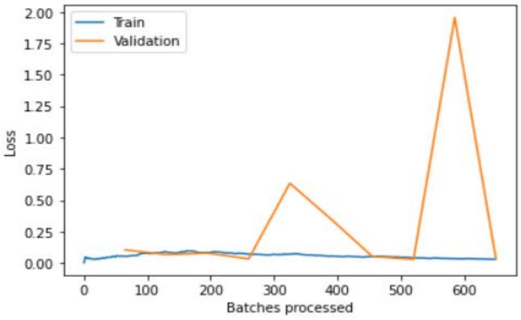
```
learn.unfreeze()
learn.lr_find()
learn.recorder.plot()
learn.fit_one_cycle(10,slice(1e-4,1e-3))
```

```
0.00% [0/2 00:00<00:00]

epoch  train_loss  valid_loss  accuracy  time

76.92% [50/65 02:22<00:42 0.0662]

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

epoch	train_loss	valid_loss	accuracy	time
0	0.053956	0.104772	0.971237	03:42
1	0.088252	0.066770	0.972196	03:48
2	0.082065	0.078531	0.970278	03:49
3	0.071285	0.032492	0.985618	03:49
4	0.070751	0.635056	0.785235	03:48
5	0.053112	0.348804	0.957814	03:51
6	0.050586	0.051793	0.979866	03:50
7	0.040913	0.029038	0.988495	03:46
8	0.035236	1.955595	0.976989	03:48
9	0.029249	0.036170	0.984660	03:53

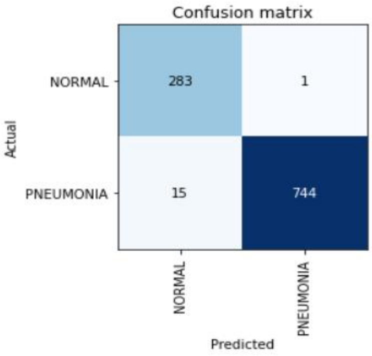


```
In [12]:
learn.recorder.plot_losses()
```



Results :

```
In [13]:
interp = ClassificationInterpretation.from_learner(learn)
interp.plot_confusion_matrix()
```



	NORMAL	PNEUMONIA
NORMAL	283	1
PNEUMONIA	15	744

Prediction Using Trained Model :

```
In [14]:
img = open_image('../input/chest-xray-pneumonia/chest_xray/test/NORMAL/IM-0001-0001.jpeg')
print(learn.predict(img)[0])
```

```
NORMAL
```