Note

之前的結果

batch_size = 32 ， image_size = 224 ， epoch = 8

	測試結果		測試結果
• resnet34	0.744375	* densenet121	0.771875
• resnet50	0.753125	* densenet169	0.785
• resnet101	0.76375	* densenet201	0.794375
• resnet152	0.764375		

調整batch_size , image_size

batch_size = 8 ， image_size = 448 ， epoch = 8

	valid_loss	accuracy	recall
• densenet201	0.470928	0.794692	0.797879

batch_size = 64 ， image_size = 112 ， epoch =15

	valid_loss	accuracy	recall
• densenet201	0.552136	0.782201	0.784134

雙模型

先分C再分 A

	測試結果
• densnet201	0.79

In [1]:

```
%reload_ext autoreload
%autoreload 2
%matplotlib inline
```

In [2]:

```
import seaborn as sns
import numpy as np
```

In [3]:

```
from fastai import *
from fastai.vision import *
from fastai.vision.learner import cnn_config
from fastai.callbacks import *
import matplotlib.pyplot as plt
from PIL import Image
import pandas as pd
import cv2
import shutil
import math
from math import floor
from fastai2.test_utils import *
```

In [4]:

```
import torch
torch.__version__
```

Out[4]:

```
'1.5.1+cu101'
```

In [5]:

```
torch.cuda.is_available()
```

Out[5]:

```
True
```

In [6]:

```
#!pip freeze > requirements.txt
```

In [7]:

```
!pip list
```

Package	Version
absl-py	0.8.0
adal	1.2.2
ansiwrap	0.8.4
applicationinsights	0.11.9
astor	0.8.0
attrs	19.3.0
azure-common	1.1.23
azure-graphrbac	0.61.1
azure-mgmt-authorization	0.60.0
azure-mgmt-containerregistry	2.8.0
azure-mgmt-keyvault	2.0.0
azure-mgmt-resource	5.1.0
azure-mgmt-storage	6.0.0
azure-storage-blob	2.1.0
azure-storage-common	2.1.0
azureml-automl-core	1.0.83.1
azureml-automl-runtime	1.0.83
azureml-contrib-notebook	1.0.83
azureml-contrib-pipeline-steps	1.0.83
azureml-contrib-server	1.0.83
azureml-contrib-services	1.0.83
azureml-core	1.0.83
azureml-dataprep	1.1.35
azureml-dataprep-native	13.2.0
azureml-defaults	1.0.83
azureml-explain-model	1.0.83
azureml-interpret	1.0.83
azureml-model-management-sdk	1.0.1b6.post1
azureml-opendatasets	1.0.83
azureml-pipeline	1.0.83
azureml-pipeline-core	1.0.83
azureml-pipeline-steps	1.0.83
azureml-sdk	1.0.83
azureml-telemetry	1.0.83
azureml-train	1.0.83
azureml-train-automl	1.0.83
azureml-train-automl-client	1.0.83
azureml-train-automl-runtime	1.0.83.1
azureml-train-core	1.0.83
azureml-train-restclients-hyperdrive	1.0.83
azureml-widgets	1.0.83
backcall	0.1.0
backports.tempfile	1.0
backports.weakref	1.0.post1
beautifulsoup4	4.9.1
bleach	3.1.0
blis	0.4.1
boto	2.49.0
boto3	1.10.10
botocore	1.13.10
Bottleneck	1.3.2
catalogue	1.0.0
certifi	2019.9.11
cffi	1.13.2
chainer	6.5.0
chardet	3.0.4
Click	7.0
cloudpickle	1.2.2
configparser	3.7.4

contextlib2	0.6.0.post1
cryptography	2.8
cycler	0.10.0
cymem	2.0.3
Cython	0.29.13
dataclasses	0.7
decorator	4.4.1
defusedxml	0.6.0
dill	0.3.1.1
distro	1.4.0
docker	4.1.0
docutils	0.15.2
dotnetcore2	2.1.10
EasyProcess	0.3
efficientnet-pytorch	0.6.3
entrypoint2	0.2.1
entrypoints	0.3
fastai	1.0.61
fastai2	0.0.18
fastcore	0.1.18
fastprogress	0.2.3
filelock	3.0.12
fire	0.2.1
flake8	3.7.9
Flask	1.1.1
fusepy	3.0.1
future	0.18.2
gast	0.2.2
gensim	3.8.1
google-pasta	0.1.7
grpcio	1.16.1
gunicorn	19.9.0
h5py	2.9.0
horovod	0.16.4
idna	2.8
imageio	2.6.1
importlib-metadata	0.23
interpret-community	0.3.1
interpret-core	0.1.19
ipykernel	5.1.3
ipython	7.9.0
ipython-genutils	0.2.0
ipywidgets	7.5.1
isodate	0.6.0
itsdangerous	1.1.0
jedi	0.15.1
jeepney	0.4.1
Jinja2	2.10.3
jmespath	0.9.4
joblib	0.13.2
json-logging-py	0.2
JsonForm	0.0.2
jsonpickle	1.2
jsonschema	3.1.1
JsonSir	0.0.2
jupyter-client	5.3.4
jupyter-core	4.6.0
Keras-Applications	1.0.8
Keras-Preprocessing	1.1.0
keras2onnx	1.6.0
kiwisolver	1.1.0

liac-arff	2.4.0
lightgbm	2.3.0
llvmlite	0.33.0
Markdown	3.1.1
MarkupSafe	1.1.1
matplotlib	3.1.1
mccabe	0.6.1
mistune	0.8.4
mkl-fft	1.0.14
mkl-random	1.1.0
mkl-service	2.3.0
more-itertools	7.2.0
msgpack	0.6.2
msrest	0.6.10
msrestazure	0.6.2
multimethods	1.0.0
munch	2.5.0
murmurhash	1.0.2
nb-conda	2.2.1
nb-conda-kernels	2.2.2
nbconvert	5.6.0
nbformat	4.4.0
ndg-httpsclient	0.5.1
networkx	2.4
nimbusml	1.5.0
notebook	6.0.1
numba	0.50.1
numexpr	2.7.1
numpy	1.19.0
nvidia-ml-py3	7.352.0
oauthlib	3.1.0
olefile	0.46
onnx	1.6.0
onnx-chainer	1.5.0
onnxconverter-common	1.6.0
onnxmltools	1.4.1
onnxruntime	1.0.0
opencv-python	4.3.0.36
opt-einsum	3.1.0
packaging	19.2
pandas	1.0.5
pandocfilters	1.4.2
papermill	1.2.1
parso	0.5.1
pathspec	0.6.0
patsy	0.5.1
pexpect	4.7.0
pickleshare	0.7.5
Pillow	6.2.1
pip	19.3.1
plac	1.1.3
pmdarima	1.1.1
preshed	3.0.2
pretrainedmodels	0.7.4
prometheus-client	0.7.1
prompt-toolkit	2.0.10
protobuf	3.9.2
psutil	5.6.4
ptyprocess	0.6.0
py-cpuinfo	5.0.0
py4j	0.10.7

pyarrow	0.15.1
pyasn1	0.4.7
pycodestyle	2.5.0
pycparser	2.19
pyflakes	2.1.1
Pygments	2.4.2
PyJWT	1.7.1
pyOpenSSL	19.0.0
pyparsing	2.4.2
pyrsistent	0.15.4
pyspark	2.4.4
python-dateutil	2.8.0
Python-EasyConfig	0.1.7
pytz	2019.3
pyunpack	0.2.1
PyWavelets	1.1.1
PyYAML	5.1.2
pyzmq	18.1.0
requests	2.22.0
requests-oauthlib	1.2.0
Resource	0.2.1
ruamel.yaml	0.15.89
s3transfer	0.2.1
scikit-image	0.16.2
scikit-learn	0.23.1
scipy	1.1.0
seaborn	0.10.1
SecretStorage	3.1.1
Send2Trash	1.5.0
setuptools	41.6.0.post20191030
shap	0.29.3
six	1.12.0
skl2onnx	1.4.9
sklearn-pandas	1.7.0
sktime	0.4.0
smart-open	1.9.0
soupsieve	2.0.1
spacy	2.3.0
srsly	1.0.2
statsmodels	0.11.1
tenacity	5.1.5
tensorboard	2.0.0
tensorflow-estimator	2.0.0
tensorflow-gpu	2.0.0
termcolor	1.1.0
terminado	0.8.2
testpath	0.4.2
textwrap3	0.9.2
tf2onnx	1.5.3
thinc	7.4.1
threadpoolctl	2.1.0
torch	1.5.1+cu101
torchvision	0.6.1+cu101
tornado	6.0.3
tqdm	4.47.0
traitlets	4.3.3
tsai	0.1.0
typing	3.6.6
typing-extensions	3.7.4.1
urllib3	1.25.6
wasabi	0.7.0

```
wcwidth                                 0.1.7
webencodings                            0.5.1
websocket-client                        0.56.0
Werkzeug                                0.16.0
wheel                                   0.30.0
widgetsnbextension                      3.5.1
wrapt                                   1.11.2
zipp                                    0.6.0
```

In [8]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [9]:

```
import zipfile
import os
def zipimage(data_name):
    cwd=os.getcwd()
    filename = data_name
    filepath = os.path.join(cwd, filename)
    with zipfile.ZipFile(filepath, 'r') as zip:
        zip.extractall()
```

In [10]:

```
#zipimage("C1-P1_Test.zip")
#zipimage("mango.zip")
```

EDA

In [11]:

```
df=pd.read_csv('train.csv')
df1=pd.read_csv('dev.csv')
```

In [12]:

```
df2=pd.concat([df,df1], ignore_index=True)
```

In [13]:

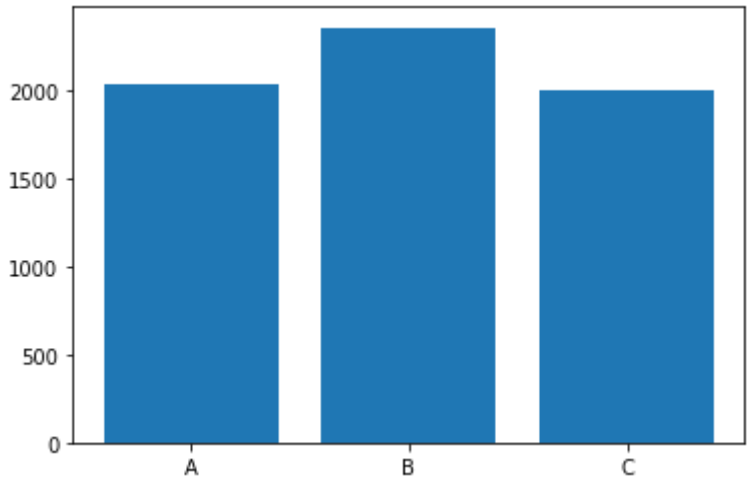
```
countA=len(df2['label'][df2['label']=='A'])
countB=len(df2['label'][df2['label']=='B'])
countC=len(df2['label'][df2['label']=='C'])
```

In [14]:

```
x=[countA,countB,countC]
plt.bar(['A','B','C'],height=x)
```

Out[14]:

```
<BarContainer object of 3 artists>
```



In [15]:

```
def concate_train_valid(imgfile:str,foldername:str):
    path=os.getcwd()+'/'+imgfile
    for ii in os.listdir(path):
        if ii!='.ipynb_checkpoints':
            shutil.copyfile(path+'/'+ii,os.getcwd()+'/'+foldername+'/'+ii)
```

In [16]:

```
"""
os.mkdir(os.getcwd()+'/train_images')
concate_train_valid('C1-P1_Train','train_images')
concate_train_valid('C1-P1_Dev','train_images')
"""
```

Out[16]:

```
"\nos.mkdir(os.getcwd()+'/train_images')\nconcate_train_valid('C1-P1
_Train','train_images')\nconcate_train_valid('C1-P1_Dev','train_imag
es')\n"
```

In [17]:

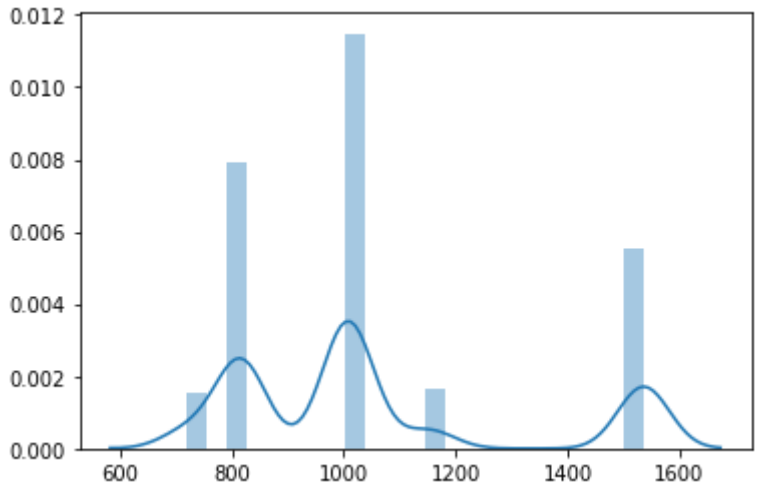
```
img_height=[]
img_width=[]
def img_stat():
    path=os.getcwd()+"/train_images"
    for ii in os.listdir(path):
        img=cv2.imread(path+'/'+ii)
        if type(img)!=type(None):
            img_height.append(img.shape[0])
            img_width.append(img.shape[1])
img_stat()
```

In [18]:

```
sns.distplot(img_height)
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f809814c4a8>
```

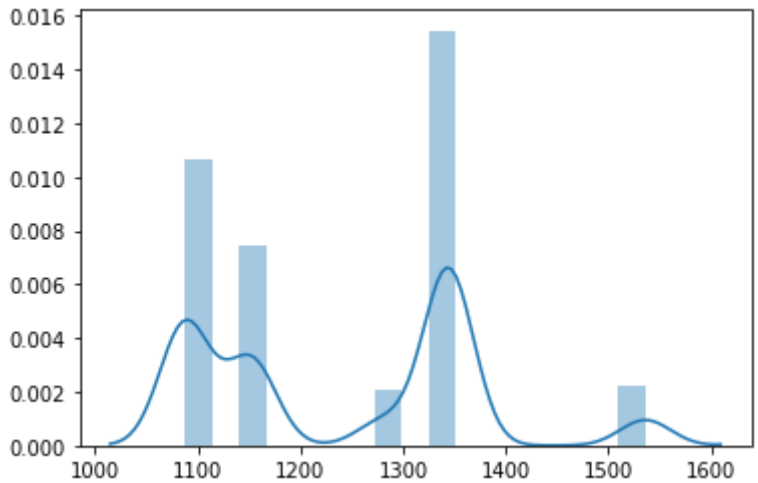


In [19]:

```
sns.distplot(img_width)
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8097ebbcc0>
```



In [20]:

```
print(max(img_height))
print(min(img_height))
print(np.mean(img_height))
print('\n')
print(max(img_width))
print(min(img_width))
print(np.mean(img_width))
```

```
1536
720
1050.5925


1536
1088
1241.71
```

In [21]:

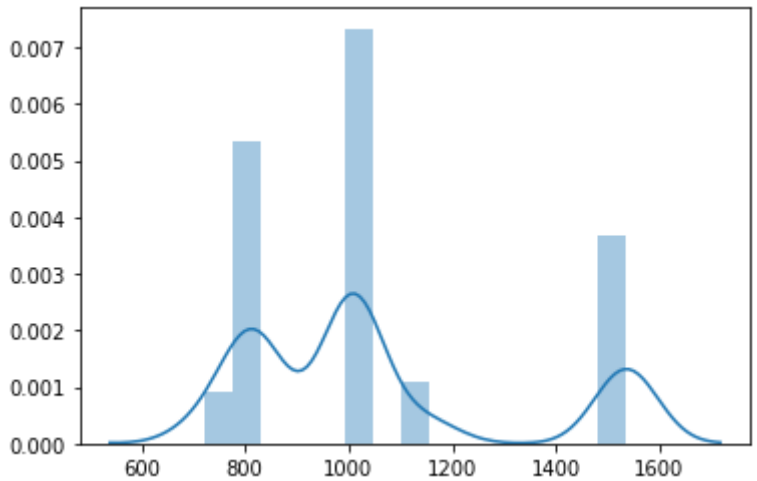
```
img_height=[]
img_width=[]
def img_stat():
    path=os.getcwd()+"/C1-P1_Test"
    for ii in os.listdir(path):
        img=cv2.imread(path+'/'+ii)
        img_height.append(img.shape[0])
        img_width.append(img.shape[1])
img_stat()
```

In [22]:

```
sns.distplot(img_height)
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8097e4b978>
```

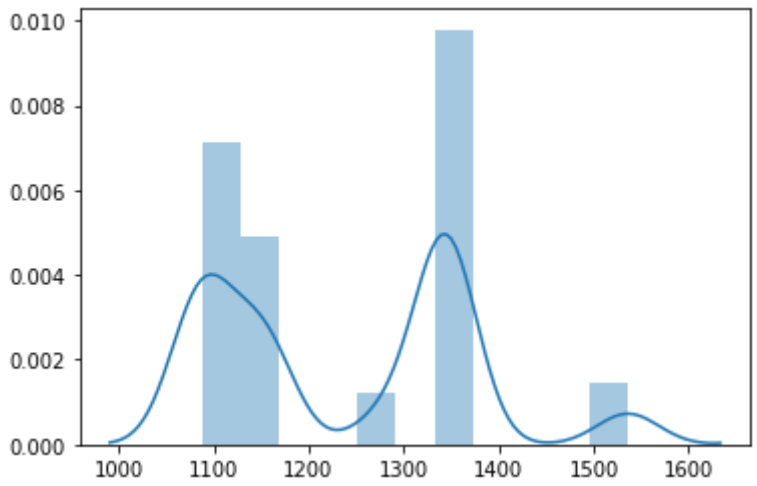


In [23]:

```
sns.distplot(img_width)
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8097e3e780>
```



In [24]:

```
print(max(img_height))
print(min(img_height))
print(np.mean(img_height))
print('\n')
print(max(img_width))
print(min(img_width))
print(np.mean(img_width))
```

```
1536
720
1052.1


1536
1088
1239.24
```

設置資料集

In [25]:

```
def image_to_folder(imagefile:str,foldername:str):
    path=os.getcwd()+'/'+imagefile
    if imagefile=="C1-P1_Dev":
        df=pd.read_csv("dev.csv")
    elif imagefile=="C1-P1_Train":
        df=pd.read_csv("train.csv")
    for i,j in enumerate(df["label"]):
            if j=="A":
                shutil.copyfile(path+'/'+df['image_id'][i],os.getcwd()+'/'+folde
rname+'A'+'/'+df['image_id'][i])
            if j=="B":
                shutil.copyfile(path+'/'+df['image_id'][i],os.getcwd()+'/'+folde
rname+'B'+'/'+df['image_id'][i])
            if j=="C":
                shutil.copyfile(path+'/'+df['image_id'][i],os.getcwd()+'/'+folde
rname+'C'+'/'+df['image_id'][i])
```

In [26]:

```
"""
os.mkdir(os.getcwd()+'/train_valid_testAB/valid')
os.mkdir(os.getcwd()+'/train_valid_testAB/train')
os.mkdir(os.getcwd()+'/train_valid_testAB')
os.mkdir(os.getcwd()+'/train_valid_test')
os.mkdir(os.getcwd()+'/train_valid_test'+'/train')
os.mkdir(os.getcwd()+'/train_valid_test'+'/train/im_A')
os.mkdir(os.getcwd()+'/train_valid_test'+'/train/im_B')
os.mkdir(os.getcwd()+'/train_valid_test'+'/train/im_C')
os.mkdir(os.getcwd()+'/train_valid_test'+'/valid')
os.mkdir(os.getcwd()+'/train_valid_test'+'/valid/im_A')
os.mkdir(os.getcwd()+'/train_valid_test'+'/valid/im_B')
os.mkdir(os.getcwd()+'/train_valid_test'+'/valid/im_C')
os.mkdir(os.getcwd()+'/train_valid_test'+'/test')
os.mkdir(os.getcwd()+'/image_C')
os.mkdir(os.getcwd()+'/image_B')
os.mkdir(os.getcwd()+'/image_A')
"""
```

Out[26]:

```
"\nos.mkdir(os.getcwd()+'/train_valid_testAB/valid')\nos.mkdir(os.ge
tcwd()+'/train_valid_testAB/train')\nos.mkdir(os.getcwd()+'/train_va
lid_testAB')\nos.mkdir(os.getcwd()+'/train_valid_test')\nos.mkdir(o
s.getcwd()+'/train_valid_test'+'/train')\nos.mkdir(os.getcwd()+'/tra
in_valid_test'+'/train/im_A')\nos.mkdir(os.getcwd()+'/train_valid_te
st'+'/train/im_B')\nos.mkdir(os.getcwd()+'/train_valid_test'+'/trai
n/im_C')\nos.mkdir(os.getcwd()+'/train_valid_test'+'/valid')\nos.mkd
ir(os.getcwd()+'/train_valid_test'+'/valid/im_A')\nos.mkdir(os.getcw
d()+'/train_valid_test'+'/valid/im_B')\nos.mkdir(os.getcwd()+'/train
_valid_test'+'/valid/im_C')\nos.mkdir(os.getcwd()+'/train_valid_tes
t'+'/test')\nos.mkdir(os.getcwd()+'/image_C')\nos.mkdir(os.getcwd()
+'/image_B')\nos.mkdir(os.getcwd()+'/image_A')\n"
```

In [27]:

```
#image_to_folder("C1-P1_Dev","image_")
#image_to_folder("C1-P1_Train","image_")
```

In [28]:

```
pathA=os.getcwd()+"/image_A"
pathB=os.getcwd()+"/image_B"
pathC=os.getcwd()+"/image_C"
```

In [29]:

```
def gen_train_valid(origin,destination:str):
    np.random.seed(42)
    indicies=np.arange(len(os.listdir(origin)))
    np.random.shuffle(indicies)
    for ii in list(indicies)[:floor(len(os.listdir(origin))*0.8)]:
        if not os.listdir(origin)[ii].endswith('jpg'):
            continue
        img = cv2.imread(origin+'/'+os.listdir(origin)[ii])
        if img.shape[0]<img.shape[1]:
            img=np.rot90(img)
        img=cv2.resize(img, (224, 224), interpolation=cv2.INTER_CUBIC)
        path = os.getcwd()+"/train_valid_test/train/"+destination
        cv2.imwrite(path+'/'+os.listdir(origin)[ii].replace('.jpg',"")+origin[-1
]+".jpg", img, [cv2.IMWRITE_JPEG_QUALITY, 100])
    for ii in list(indicies)[floor(len(os.listdir(origin))*0.8):]:
        if not os.listdir(origin)[ii].endswith('jpg'):
            continue
        img = cv2.imread(origin+'/'+os.listdir(origin)[ii])
        if img.shape[0]<img.shape[1]:
            img=np.rot90(img)
        path = os.getcwd()+"/train_valid_test/valid/"+destination
        img=cv2.resize(img, (224, 224), interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(path+'/'+os.listdir(origin)[ii].replace('.jpg',"")+origin[-1
]+".jpg", img, [cv2.IMWRITE_JPEG_QUALITY, 100])
```

In [30]:

```
def gen_test():
    path=os.getcwd()+"/C1-P1_Test"
    for ii in os.listdir(path):
        img=cv2.imread(path+'/'+ii)
        img=cv2.resize(img, (224, 224), interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(os.getcwd()+"/train_valid_test/test/"+ii, img, [cv2.IMWRITE_
JPEG_QUALITY, 100])
#gen_test()
```

In [31]:

```
#gen_train_valid(pathA,"im_A")
#gen_train_valid(pathB,"im_B")
#gen_train_valid(pathC,"im_C")
```

In [32]:

```
path = os.getcwd()+"/train_valid_test"
data = ImageList.from_folder(path).split_by_folder('train','valid')
```

In [33]:

```
func = lambda x: str(x)[-5]
data=data.label_from_func(func)
```

In [34]:

```
data
```

Out[34]:

```
LabelLists;

Train: LabelList (5119 items)
x: ImageList
Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image
(3, 224, 224),Image (3, 224, 224)
y: CategoryList
C,C,C,C,C
Path: /data/home/aistudent/notebooks/fastai/course-v3/nbs/dl1/train_
valid_test;

Valid: LabelList (1281 items)
x: ImageList
Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image
(3, 224, 224),Image (3, 224, 224)
y: CategoryList
C,C,C,C,C
Path: /data/home/aistudent/notebooks/fastai/course-v3/nbs/dl1/train_
valid_test;

Test: None
```

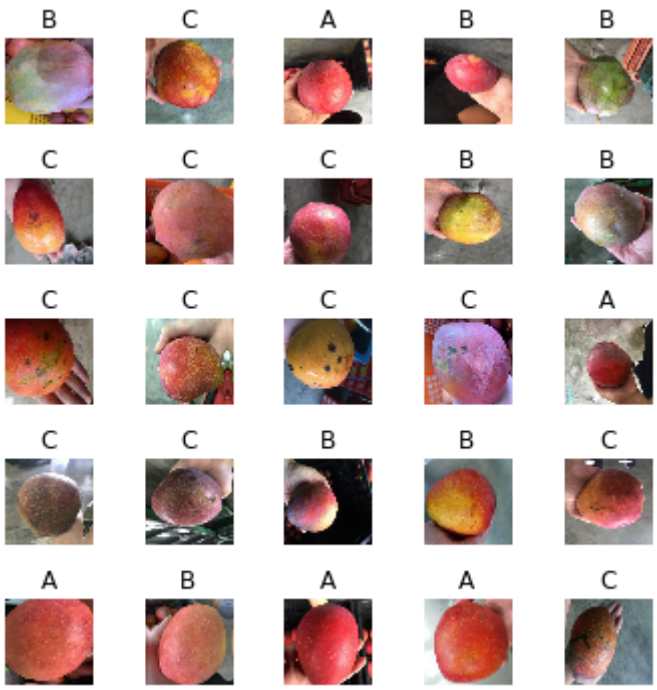
In [35]:

```
tfms = get_transforms(do_flip=True,flip_vert=True)
bs=32
data=data.transform(tfms) \
          .databunch(bs=bs) \
          .normalize(imagenet_stats)
```

In [36]:

```
data.show_batch(rows=5,figsize=(5,5))
print(f'{data.classes} {data.c}')
```

```
['A', 'B', 'C'] 3
```



Model

Resnet

In [37]:

```
arch = models.resnet34
```

In [38]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True)
```

In [39]:

```
callbacks = [
SaveModelCallback(learn, monitor='recall',mode='max',name='res34')
,ShowGraph(learn),
]
learn.callbacks = callbacks
```

In [40]:

```
learn.summary()
```

Out[40]:

```
Sequential
==========================================================================
==
Layer (type)          Output Shape          Param #    Trainable
==========================================================================
==
Conv2d                [64, 112, 112]        9,408      False
__________________________________________________________________________
__
BatchNorm2d           [64, 112, 112]        128        True
__________________________________________________________________________
__
ReLU                  [64, 112, 112]        0          False
__________________________________________________________________________
__
MaxPool2d             [64, 56, 56]          0          False
__________________________________________________________________________
__
Conv2d                [64, 56, 56]          36,864     False
__________________________________________________________________________
__
BatchNorm2d           [64, 56, 56]          128        True
__________________________________________________________________________
__
ReLU                  [64, 56, 56]          0          False
__________________________________________________________________________
__
Conv2d                [64, 56, 56]          36,864     False
__________________________________________________________________________
__
BatchNorm2d           [64, 56, 56]          128        True
__________________________________________________________________________
__
Conv2d                [64, 56, 56]          36,864     False
__________________________________________________________________________
__
BatchNorm2d           [64, 56, 56]          128        True
__________________________________________________________________________
__
ReLU                  [64, 56, 56]          0          False
__________________________________________________________________________
__
Conv2d                [64, 56, 56]          36,864     False
__________________________________________________________________________
__
BatchNorm2d           [64, 56, 56]          128        True
__________________________________________________________________________
__
Conv2d                [64, 56, 56]          36,864     False
__________________________________________________________________________
__
BatchNorm2d           [64, 56, 56]          128        True
__________________________________________________________________________
__
ReLU                  [64, 56, 56]          0          False
__________________________________________________________________________
__
Conv2d                [64, 56, 56]          36,864     False
__________________________________________________________________________
```

BatchNorm2d	[64, 56, 56]	128	True
Conv2d	[128, 28, 28]	73,728	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True
Conv2d	[128, 28, 28]	8,192	False
BatchNorm2d	[128, 28, 28]	256	True
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True

ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	147,456	False
BatchNorm2d	[128, 28, 28]	256	True
Conv2d	[256, 14, 14]	294,912	False
BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[256, 14, 14]	32,768	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[256, 14, 14]	589,824	False

BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[256, 14, 14]	589,824	False
BatchNorm2d	[256, 14, 14]	512	True
Conv2d	[512, 7, 7]	1,179,648	False
BatchNorm2d	[512, 7, 7]	1,024	True
ReLU	[512, 7, 7]	0	False
Conv2d	[512, 7, 7]	2,359,296	False
BatchNorm2d	[512, 7, 7]	1,024	True
Conv2d	[512, 7, 7]	131,072	False

BatchNorm2d	[512, 7, 7]	1,024	True
Conv2d	[512, 7, 7]	2,359,296	False
BatchNorm2d	[512, 7, 7]	1,024	True
ReLU	[512, 7, 7]	0	False
Conv2d	[512, 7, 7]	2,359,296	False
BatchNorm2d	[512, 7, 7]	1,024	True
Conv2d	[512, 7, 7]	2,359,296	False
BatchNorm2d	[512, 7, 7]	1,024	True
ReLU	[512, 7, 7]	0	False
Conv2d	[512, 7, 7]	2,359,296	False
BatchNorm2d	[512, 7, 7]	1,024	True
AdaptiveAvgPool2d	[512, 1, 1]	0	False
AdaptiveMaxPool2d	[512, 1, 1]	0	False
Flatten	[1024]	0	False
BatchNorm1d	[1024]	2,048	True
Dropout	[1024]	0	False
Linear	[512]	524,800	True
ReLU	[512]	0	False
BatchNorm1d	[512]	1,024	True
Dropout	[512]	0	False

```
Linear                [3]                  1,539      True
____________________________________________________________________________
__

Total params: 21,814,083
Total trainable params: 546,435
Total non-trainable params: 21,267,648
Optimized with 'torch.optim.adam.Adam', betas=(0.9, 0.99)
Using true weight decay as discussed in https://www.fast.ai/2018/07/
02/adam-weight-decay/
Loss function : LabelSmoothingCrossEntropy
============================================================================
==
Callbacks functions applied
    SaveModelCallback
    ShowGraph
```

In [41]:

```
learn.lr_find()
```

0.00% [0/1 00:00<00:00]

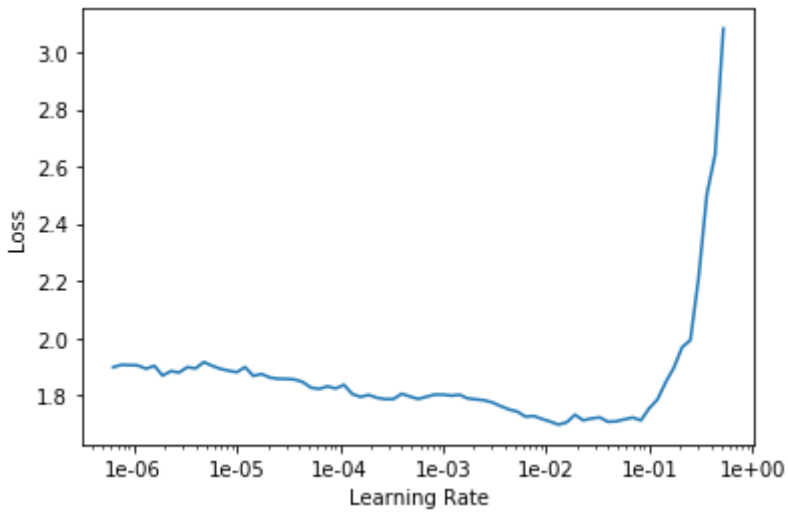
epoch	train_loss	valid_loss	accuracy	recall	f_beta	time

55.97% [89/159 00:22<00:17 6.1494]

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

In [42]:

```
learn.recorder.plot()
```

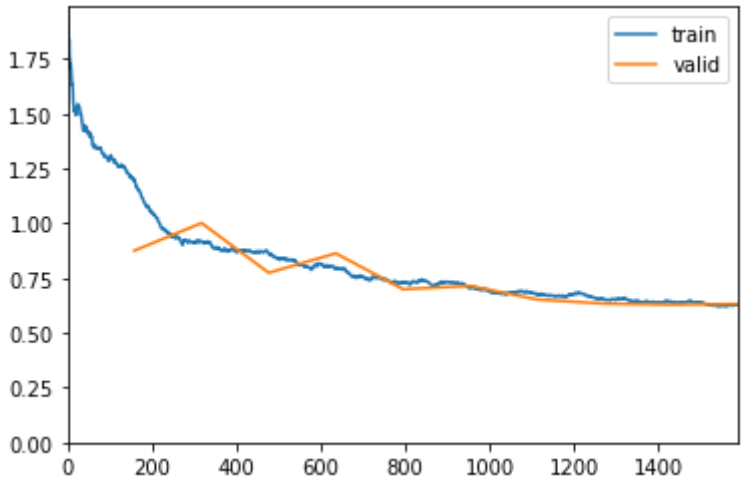


In [43]:

```
#first train
learn.fit_one_cycle(10,max_lr=1e-02)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	1.199613	0.875169	0.702576	0.702576	0.708000	00:43
1	0.917239	1.000702	0.654176	0.654176	0.650533	00:43
2	0.866330	0.773982	0.733802	0.733802	0.731310	00:43
3	0.796399	0.862471	0.665886	0.665886	0.661456	00:43
4	0.726580	0.698858	0.755660	0.755660	0.758542	00:43
5	0.702070	0.713912	0.761124	0.761124	0.765300	00:43
6	0.677057	0.651900	0.786885	0.786885	0.791700	00:43
7	0.653699	0.633098	0.797814	0.797814	0.802216	00:43
8	0.639181	0.628710	0.794692	0.794692	0.799255	00:43
9	0.629452	0.631898	0.790788	0.790788	0.795573	00:43

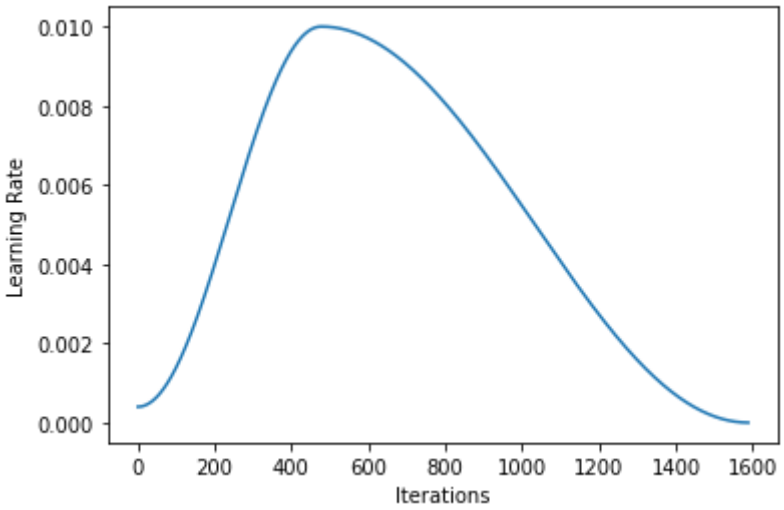
```
Better model found at epoch 0 with recall value: 0.7025761008262634.
```




```
Better model found at epoch 2 with recall value: 0.7338017225265503.
Better model found at epoch 4 with recall value: 0.7556595802307129.
Better model found at epoch 5 with recall value: 0.7611241340637207.
Better model found at epoch 6 with recall value: 0.7868852615356445.
Better model found at epoch 7 with recall value: 0.7978142499923706.
```

In [44]:

```
learn.recorder.plot_lr()
```

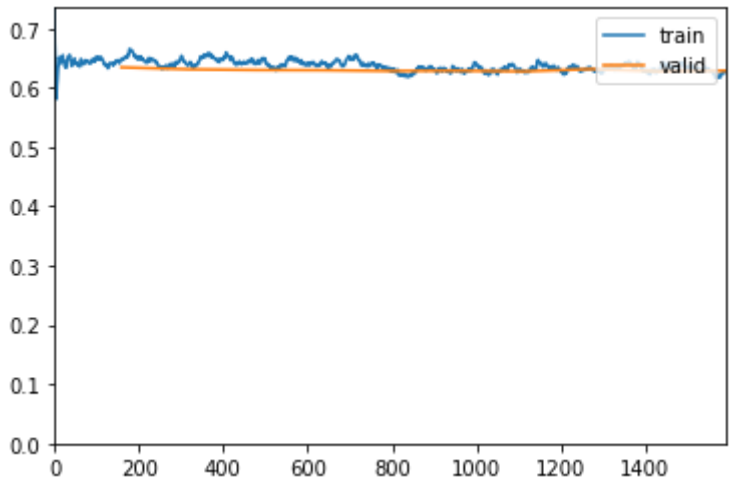


In [45]:

```
#fine tune
learn.fit_one_cycle(10,max_lr=1e-04)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	0.647781	0.634375	0.793130	0.793130	0.797187	00:43
1	0.638372	0.631064	0.793911	0.793911	0.798350	00:43
2	0.639754	0.629907	0.793130	0.793130	0.797673	00:43
3	0.643445	0.629543	0.797034	0.797034	0.801413	00:43
4	0.629314	0.628218	0.797814	0.797814	0.802112	00:43
5	0.629982	0.628618	0.791569	0.791569	0.795862	00:43
6	0.626439	0.627570	0.793911	0.793911	0.798597	00:43
7	0.632871	0.631505	0.792350	0.792350	0.797271	00:43
8	0.628762	0.627669	0.792350	0.792350	0.797015	00:43
9	0.626923	0.628697	0.785324	0.785324	0.790102	00:43

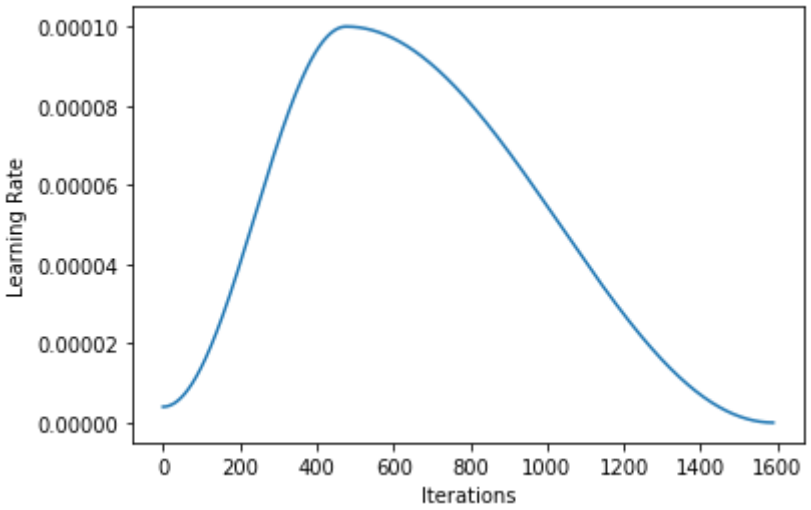
```
Better model found at epoch 0 with recall value: 0.7931303381919861.
```




```
Better model found at epoch 1 with recall value: 0.7939109802246094.
Better model found at epoch 3 with recall value: 0.7970335483551025.
Better model found at epoch 4 with recall value: 0.7978142499923706.
```

In [46]:

```
learn.recorder.plot_lr()
```



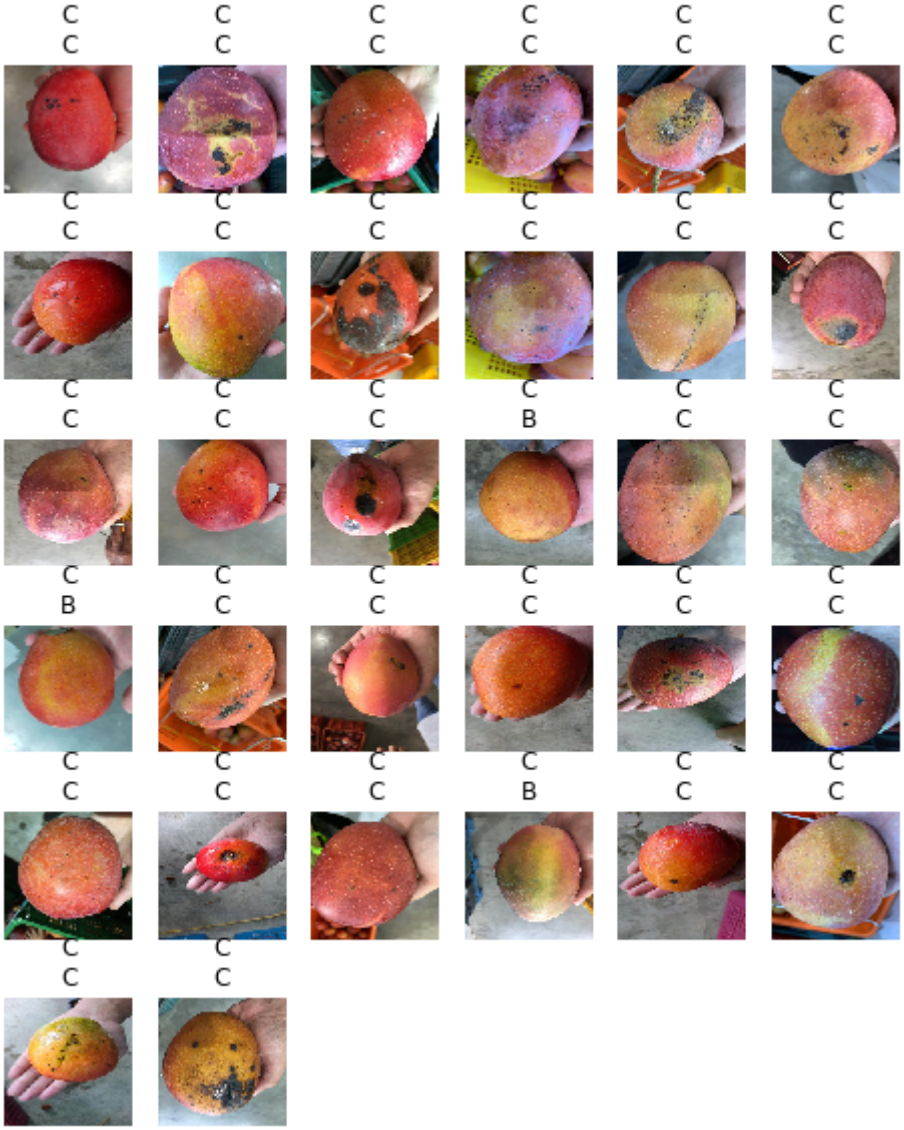
In [47]:

```
learn.save('resnet34_10_epochs')
```

In [48]:

```
learn.show_results(ds_type=DatasetType.Valid, rows=6, figsize=(8,10))
```

Ground truth
Predictions

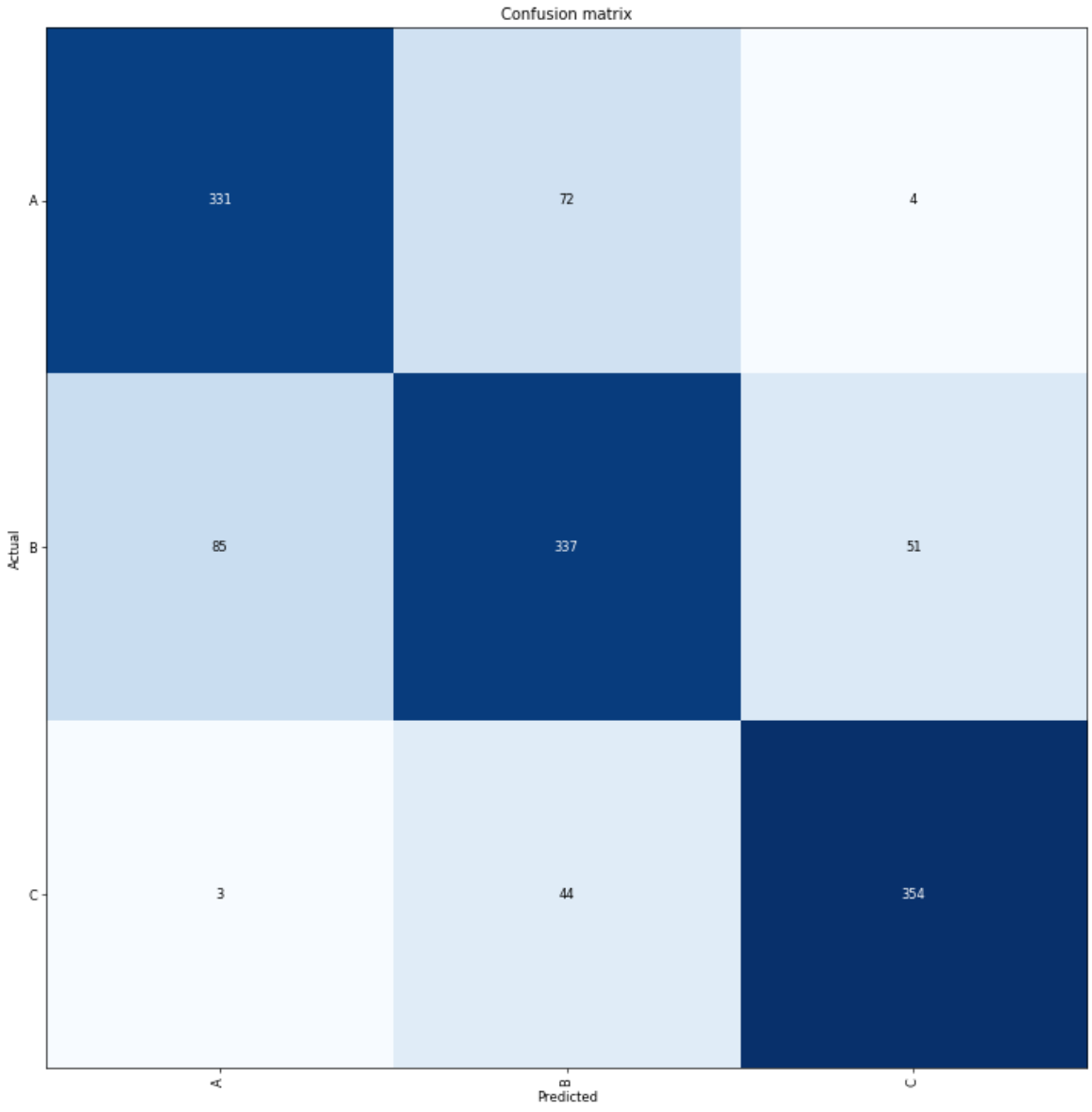


In [49]:

```
interp=ClassificationInterpretation.from_learner(learn)
```

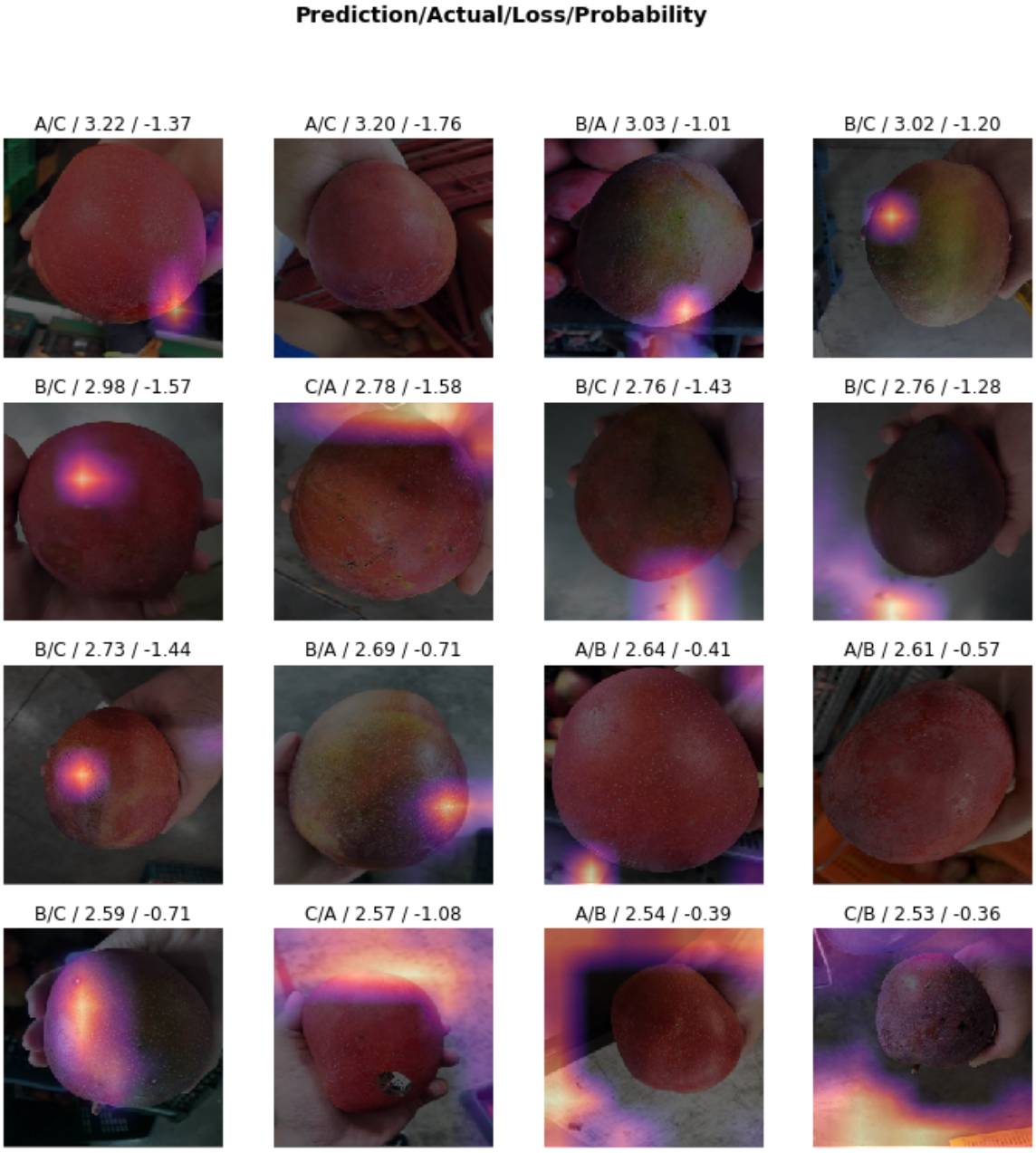
In [50]:

```
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```



In [51]:

```
interp.plot_top_losses(16,heatmap=True)
```



Densenet

In [52]:

```
arch = models.densenet121
```

In [53]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True)
```

In [54]:

```
callbacks = [
SaveModelCallback(learn, monitor='recall',mode='max',name='densenet121')
,ShowGraph(learn),
]
learn.callbacks = callbacks
```

In [55]:

```
learn.summary()
```

Out[55]:

```
Sequential
==========================================================================
==
Layer (type)          Output Shape          Param #    Trainable
==========================================================================
==
Conv2d                [64, 112, 112]        9,408      False
--------------------------------------------------------------------------
--
BatchNorm2d           [64, 112, 112]        128        True
--------------------------------------------------------------------------
--
ReLU                  [64, 112, 112]        0          False
--------------------------------------------------------------------------
--
MaxPool2d             [64, 56, 56]          0          False
--------------------------------------------------------------------------
--
BatchNorm2d           [64, 56, 56]          128        True
--------------------------------------------------------------------------
--
ReLU                  [64, 56, 56]          0          False
--------------------------------------------------------------------------
--
Conv2d                [128, 56, 56]         8,192      False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
--
Conv2d                [32, 56, 56]          36,864     False
--------------------------------------------------------------------------
--
BatchNorm2d           [96, 56, 56]          192        True
--------------------------------------------------------------------------
--
ReLU                  [96, 56, 56]          0          False
--------------------------------------------------------------------------
--
Conv2d                [128, 56, 56]         12,288     False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
--
Conv2d                [32, 56, 56]          36,864     False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
```

Conv2d	[128, 56, 56]	16,384	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[160, 56, 56]	320	True
ReLU	[160, 56, 56]	0	False
Conv2d	[128, 56, 56]	20,480	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[192, 56, 56]	384	True
ReLU	[192, 56, 56]	0	False
Conv2d	[128, 56, 56]	24,576	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[224, 56, 56]	448	True
ReLU	[224, 56, 56]	0	False
Conv2d	[128, 56, 56]	28,672	False
BatchNorm2d	[128, 56, 56]	256	True

ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[256, 56, 56]	512	True
ReLU	[256, 56, 56]	0	False
Conv2d	[128, 56, 56]	32,768	False
AvgPool2d	[128, 28, 28]	0	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	16,384	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[160, 28, 28]	320	True
ReLU	[160, 28, 28]	0	False
Conv2d	[128, 28, 28]	20,480	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[192, 28, 28]	384	True
ReLU	[192, 28, 28]	0	False
Conv2d	[128, 28, 28]	24,576	False

BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[224, 28, 28]	448	True
ReLU	[224, 28, 28]	0	False
Conv2d	[128, 28, 28]	28,672	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[256, 28, 28]	512	True
ReLU	[256, 28, 28]	0	False
Conv2d	[128, 28, 28]	32,768	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[288, 28, 28]	576	True
ReLU	[288, 28, 28]	0	False
Conv2d	[128, 28, 28]	36,864	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False

Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[320, 28, 28]	640	True
ReLU	[320, 28, 28]	0	False
Conv2d	[128, 28, 28]	40,960	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[352, 28, 28]	704	True
ReLU	[352, 28, 28]	0	False
Conv2d	[128, 28, 28]	45,056	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[384, 28, 28]	768	True
ReLU	[384, 28, 28]	0	False
Conv2d	[128, 28, 28]	49,152	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[416, 28, 28]	832	True

ReLU	[416, 28, 28]	0	False
Conv2d	[128, 28, 28]	53,248	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[448, 28, 28]	896	True
ReLU	[448, 28, 28]	0	False
Conv2d	[128, 28, 28]	57,344	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[480, 28, 28]	960	True
ReLU	[480, 28, 28]	0	False
Conv2d	[128, 28, 28]	61,440	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[512, 28, 28]	1,024	True
ReLU	[512, 28, 28]	0	False
Conv2d	[256, 28, 28]	131,072	False
AvgPool2d	[256, 14, 14]	0	False

BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[128, 14, 14]	32,768	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[288, 14, 14]	576	True
ReLU	[288, 14, 14]	0	False
Conv2d	[128, 14, 14]	36,864	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[320, 14, 14]	640	True
ReLU	[320, 14, 14]	0	False
Conv2d	[128, 14, 14]	40,960	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[352, 14, 14]	704	True
ReLU	[352, 14, 14]	0	False

Conv2d	[128, 14, 14]	45,056	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[384, 14, 14]	768	True
ReLU	[384, 14, 14]	0	False
Conv2d	[128, 14, 14]	49,152	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[416, 14, 14]	832	True
ReLU	[416, 14, 14]	0	False
Conv2d	[128, 14, 14]	53,248	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[448, 14, 14]	896	True
ReLU	[448, 14, 14]	0	False
Conv2d	[128, 14, 14]	57,344	False
BatchNorm2d	[128, 14, 14]	256	True

ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[480, 14, 14]	960	True
ReLU	[480, 14, 14]	0	False
Conv2d	[128, 14, 14]	61,440	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[512, 14, 14]	1,024	True
ReLU	[512, 14, 14]	0	False
Conv2d	[128, 14, 14]	65,536	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[544, 14, 14]	1,088	True
ReLU	[544, 14, 14]	0	False
Conv2d	[128, 14, 14]	69,632	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[576, 14, 14]	1,152	True

ReLU	[576, 14, 14]	0	False
Conv2d	[128, 14, 14]	73,728	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[608, 14, 14]	1,216	True
ReLU	[608, 14, 14]	0	False
Conv2d	[128, 14, 14]	77,824	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[640, 14, 14]	1,280	True
ReLU	[640, 14, 14]	0	False
Conv2d	[128, 14, 14]	81,920	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[672, 14, 14]	1,344	True
ReLU	[672, 14, 14]	0	False
Conv2d	[128, 14, 14]	86,016	False

BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[704, 14, 14]	1,408	True
ReLU	[704, 14, 14]	0	False
Conv2d	[128, 14, 14]	90,112	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[736, 14, 14]	1,472	True
ReLU	[736, 14, 14]	0	False
Conv2d	[128, 14, 14]	94,208	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[768, 14, 14]	1,536	True
ReLU	[768, 14, 14]	0	False
Conv2d	[128, 14, 14]	98,304	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False

Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[800, 14, 14]	1,600	True
ReLU	[800, 14, 14]	0	False
Conv2d	[128, 14, 14]	102,400	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[832, 14, 14]	1,664	True
ReLU	[832, 14, 14]	0	False
Conv2d	[128, 14, 14]	106,496	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[864, 14, 14]	1,728	True
ReLU	[864, 14, 14]	0	False
Conv2d	[128, 14, 14]	110,592	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[896, 14, 14]	1,792	True
ReLU	[896, 14, 14]	0	False

Conv2d	[128, 14, 14]	114,688	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[928, 14, 14]	1,856	True
ReLU	[928, 14, 14]	0	False
Conv2d	[128, 14, 14]	118,784	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[960, 14, 14]	1,920	True
ReLU	[960, 14, 14]	0	False
Conv2d	[128, 14, 14]	122,880	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[992, 14, 14]	1,984	True
ReLU	[992, 14, 14]	0	False
Conv2d	[128, 14, 14]	126,976	False
BatchNorm2d	[128, 14, 14]	256	True

ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[1024, 14, 14]	2,048	True
ReLU	[1024, 14, 14]	0	False
Conv2d	[512, 14, 14]	524,288	False
AvgPool2d	[512, 7, 7]	0	False
BatchNorm2d	[512, 7, 7]	1,024	True
ReLU	[512, 7, 7]	0	False
Conv2d	[128, 7, 7]	65,536	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[544, 7, 7]	1,088	True
ReLU	[544, 7, 7]	0	False
Conv2d	[128, 7, 7]	69,632	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[576, 7, 7]	1,152	True
ReLU	[576, 7, 7]	0	False

Conv2d	[128, 7, 7]	73,728	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[608, 7, 7]	1,216	True
ReLU	[608, 7, 7]	0	False
Conv2d	[128, 7, 7]	77,824	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[640, 7, 7]	1,280	True
ReLU	[640, 7, 7]	0	False
Conv2d	[128, 7, 7]	81,920	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[672, 7, 7]	1,344	True
ReLU	[672, 7, 7]	0	False
Conv2d	[128, 7, 7]	86,016	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False

Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[704, 7, 7]	1,408	True
ReLU	[704, 7, 7]	0	False
Conv2d	[128, 7, 7]	90,112	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[736, 7, 7]	1,472	True
ReLU	[736, 7, 7]	0	False
Conv2d	[128, 7, 7]	94,208	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[768, 7, 7]	1,536	True
ReLU	[768, 7, 7]	0	False
Conv2d	[128, 7, 7]	98,304	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[800, 7, 7]	1,600	True

ReLU	[800, 7, 7]	0	False
Conv2d	[128, 7, 7]	102,400	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[832, 7, 7]	1,664	True
ReLU	[832, 7, 7]	0	False
Conv2d	[128, 7, 7]	106,496	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[864, 7, 7]	1,728	True
ReLU	[864, 7, 7]	0	False
Conv2d	[128, 7, 7]	110,592	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[896, 7, 7]	1,792	True
ReLU	[896, 7, 7]	0	False
Conv2d	[128, 7, 7]	114,688	False

BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[928, 7, 7]	1,856	True
ReLU	[928, 7, 7]	0	False
Conv2d	[128, 7, 7]	118,784	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[960, 7, 7]	1,920	True
ReLU	[960, 7, 7]	0	False
Conv2d	[128, 7, 7]	122,880	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[992, 7, 7]	1,984	True
ReLU	[992, 7, 7]	0	False
Conv2d	[128, 7, 7]	126,976	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False

```
__
BatchNorm2d          [1024, 7, 7]         2,048      True
__
AdaptiveAvgPool2d    [1024, 1, 1]         0          False
__
AdaptiveMaxPool2d    [1024, 1, 1]         0          False
__
Flatten              [2048]               0          False
__
BatchNorm1d          [2048]               4,096      True
__
Dropout              [2048]               0          False
__
Linear               [512]                1,049,088  True
__
ReLU                 [512]                0          False
__
BatchNorm1d          [512]                1,024      True
__
Dropout              [512]                0          False
__
Linear               [3]                  1,539      True
__

Total params: 8,009,603
Total trainable params: 1,139,395
Total non-trainable params: 6,870,208
Optimized with 'torch.optim.adam.Adam', betas=(0.9, 0.99)
Using true weight decay as discussed in https://www.fast.ai/2018/07/
02/adam-weight-decay/
Loss function : LabelSmoothingCrossEntropy
==========================================================================
==
Callbacks functions applied
    SaveModelCallback
    ShowGraph
```

In [56]:

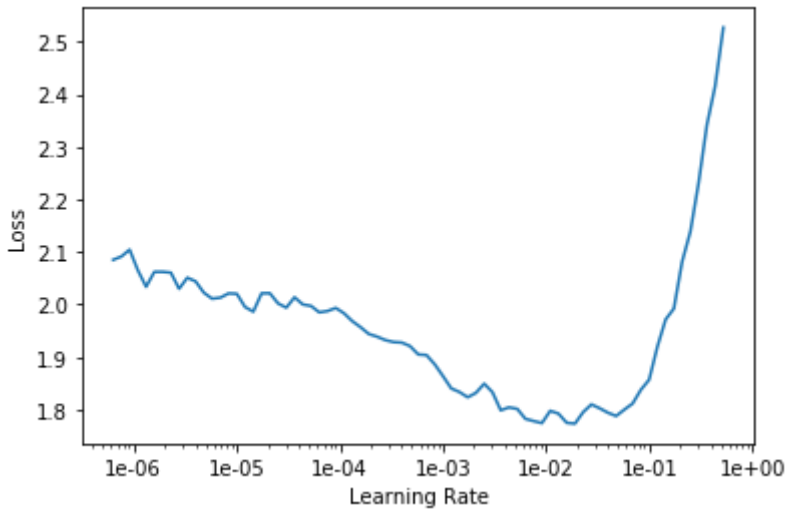
```
learn.lr_find()
learn.recorder.plot()
```

0.00% [0/1 00:00<00:00]

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time

55.97% [89/159 00:53<00:41 4.9297]

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

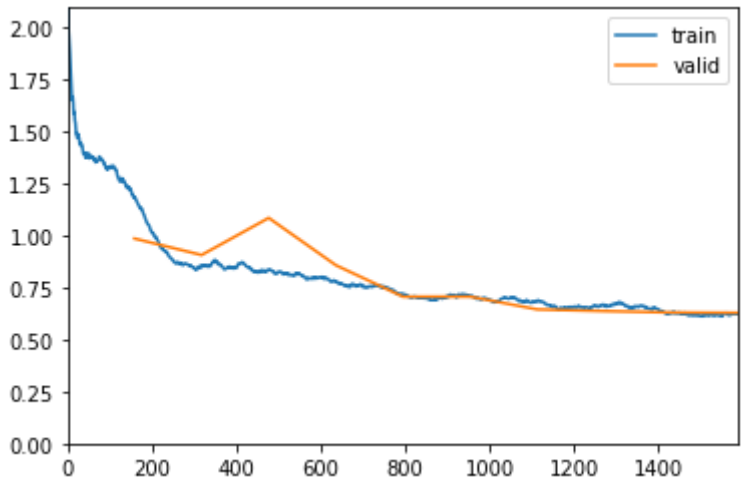


In [57]:

```
#first train
learn.fit_one_cycle(10,max_lr=1e-02)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	1.178555	0.983484	0.676034	0.676034	0.672097	01:36
1	0.852956	0.904527	0.661202	0.661202	0.661922	01:36
2	0.834430	1.082520	0.518345	0.518345	0.461860	01:36
3	0.776987	0.855435	0.634660	0.634660	0.625802	01:36
4	0.713921	0.704363	0.772053	0.772053	0.768333	01:36
5	0.707907	0.704679	0.745511	0.745511	0.749918	01:36
6	0.683843	0.643420	0.775176	0.775176	0.780165	01:36
7	0.664922	0.635343	0.794692	0.794692	0.797167	01:36
8	0.626594	0.628624	0.796253	0.796253	0.800036	01:35
9	0.623650	0.627012	0.795472	0.795472	0.799414	01:35

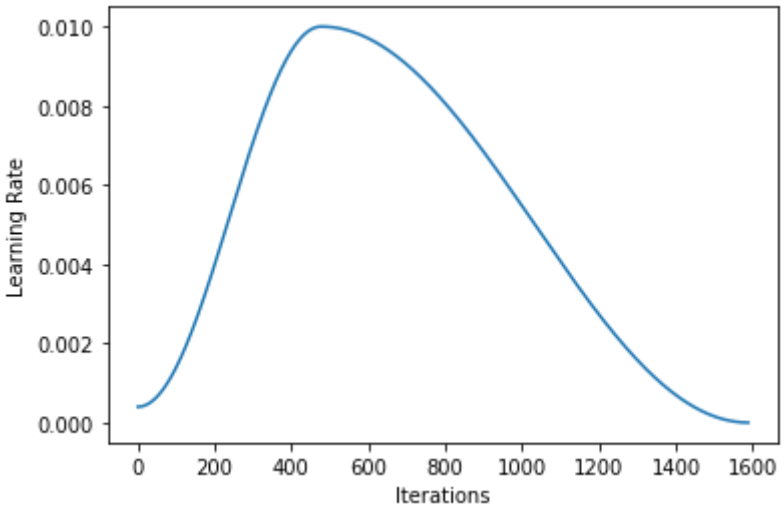
```
Better model found at epoch 0 with recall value: 0.6760343909263611.
```




```
Better model found at epoch 4 with recall value: 0.772053062915802.
Better model found at epoch 6 with recall value: 0.7751755714416504.
Better model found at epoch 7 with recall value: 0.7946916818618774.
Better model found at epoch 8 with recall value: 0.796252965927124.
```

In [58]:

```
learn.recorder.plot_lr()
```

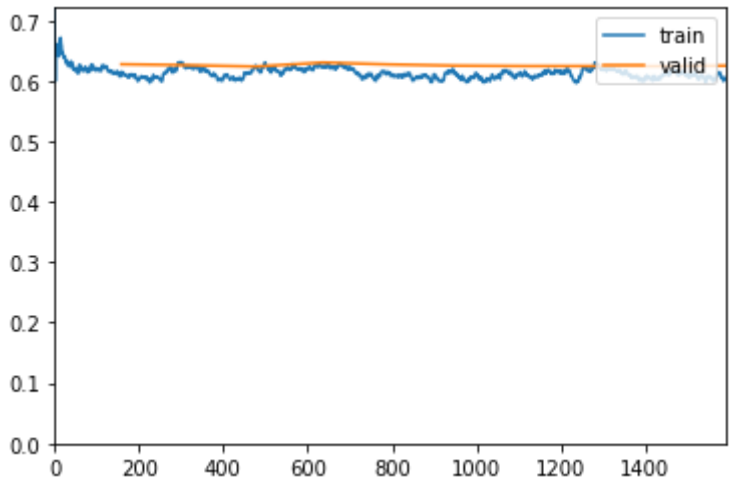


In [59]:

```
#fine tune
learn.fit_one_cycle(10,max_lr=1e-04)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	0.614471	0.627459	0.801717	0.801717	0.805035	01:36
1	0.621913	0.625893	0.799375	0.799375	0.802877	01:36
2	0.623346	0.623666	0.798595	0.798595	0.802502	01:36
3	0.622087	0.630280	0.799375	0.799375	0.802055	01:36
4	0.612734	0.627104	0.797034	0.797034	0.800469	01:36
5	0.610101	0.625086	0.797814	0.797814	0.801540	01:36
6	0.611468	0.624455	0.802498	0.802498	0.806069	01:36
7	0.616699	0.624427	0.799375	0.799376	0.802967	01:36
8	0.604135	0.624156	0.800156	0.800156	0.803872	01:36
9	0.604783	0.625001	0.802498	0.802498	0.805941	01:36

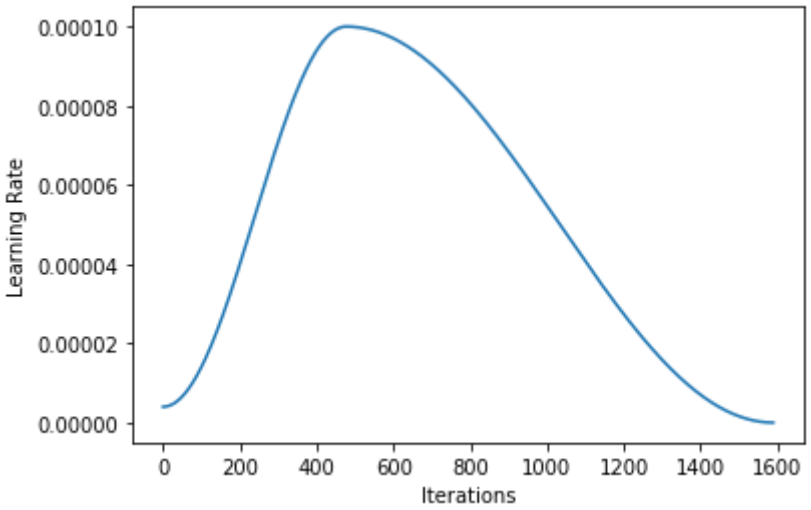
```
Better model found at epoch 0 with recall value: 0.8017174005508423.
```




```
Better model found at epoch 6 with recall value: 0.8024979829788208.
```

In [60]:

```
learn.recorder.plot_lr()
```



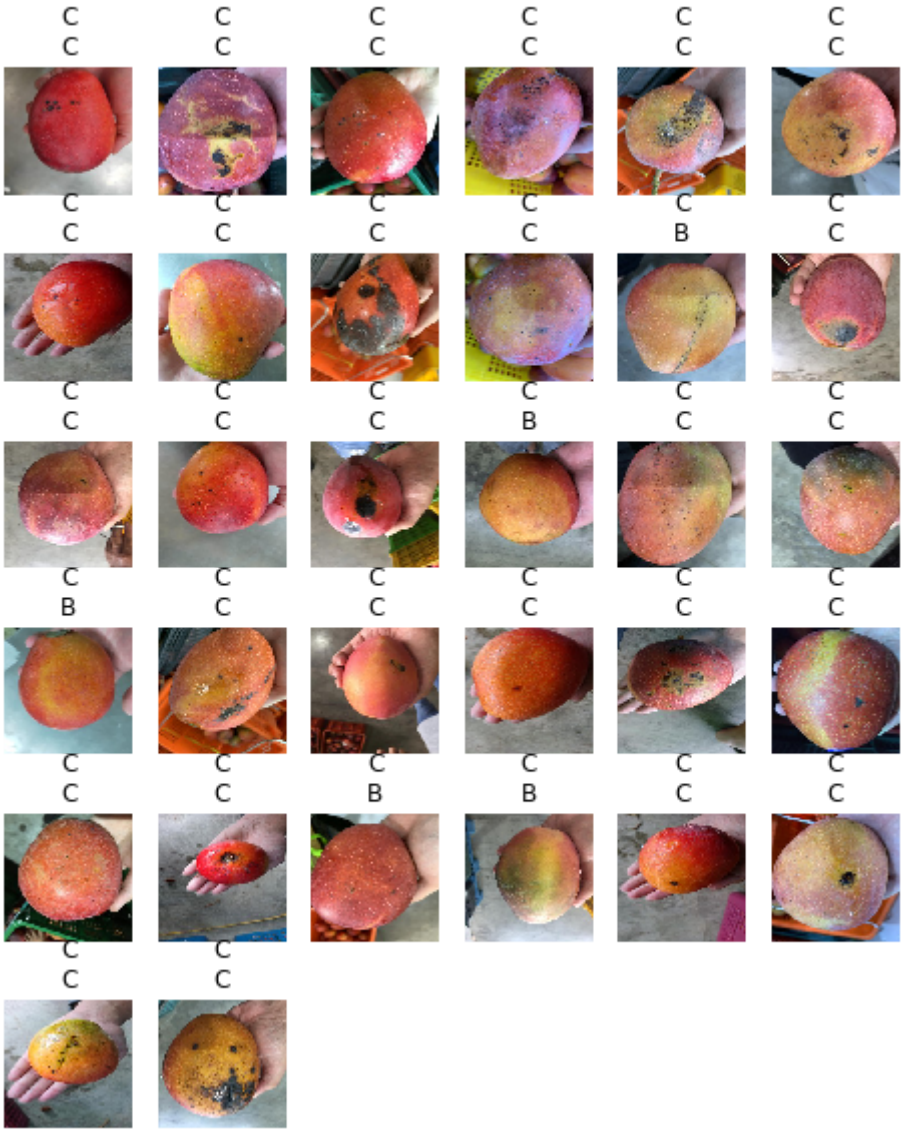
In [61]:

```
learn.save('densenet121_10_epochs')
```

In [62]:

```
learn.show_results(ds_type=DatasetType.Valid, rows=6, figsize=(8,10))
```

Ground truth
Predictions

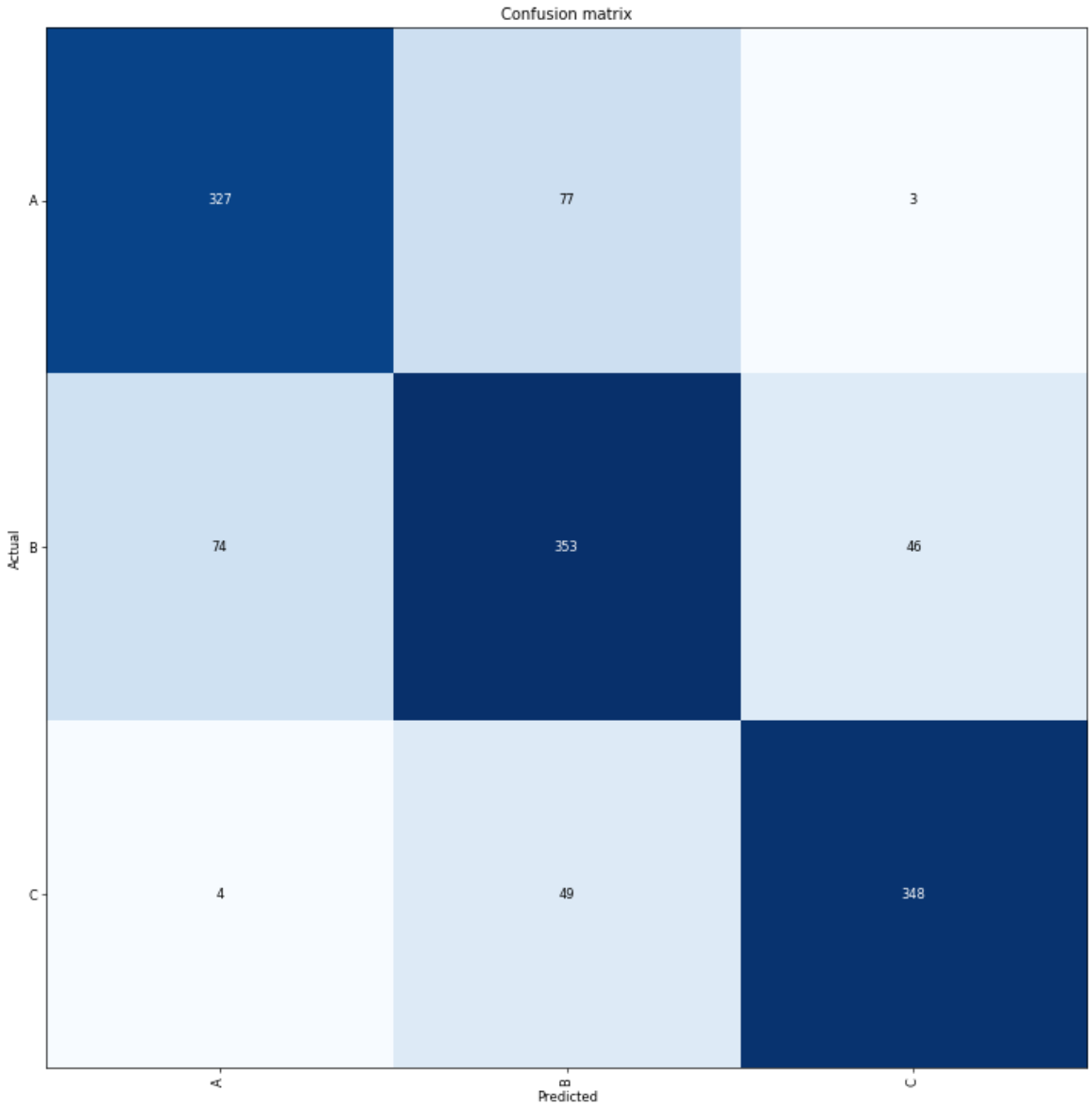


In [64]:

```
interp=ClassificationInterpretation.from_learner(learn)
```

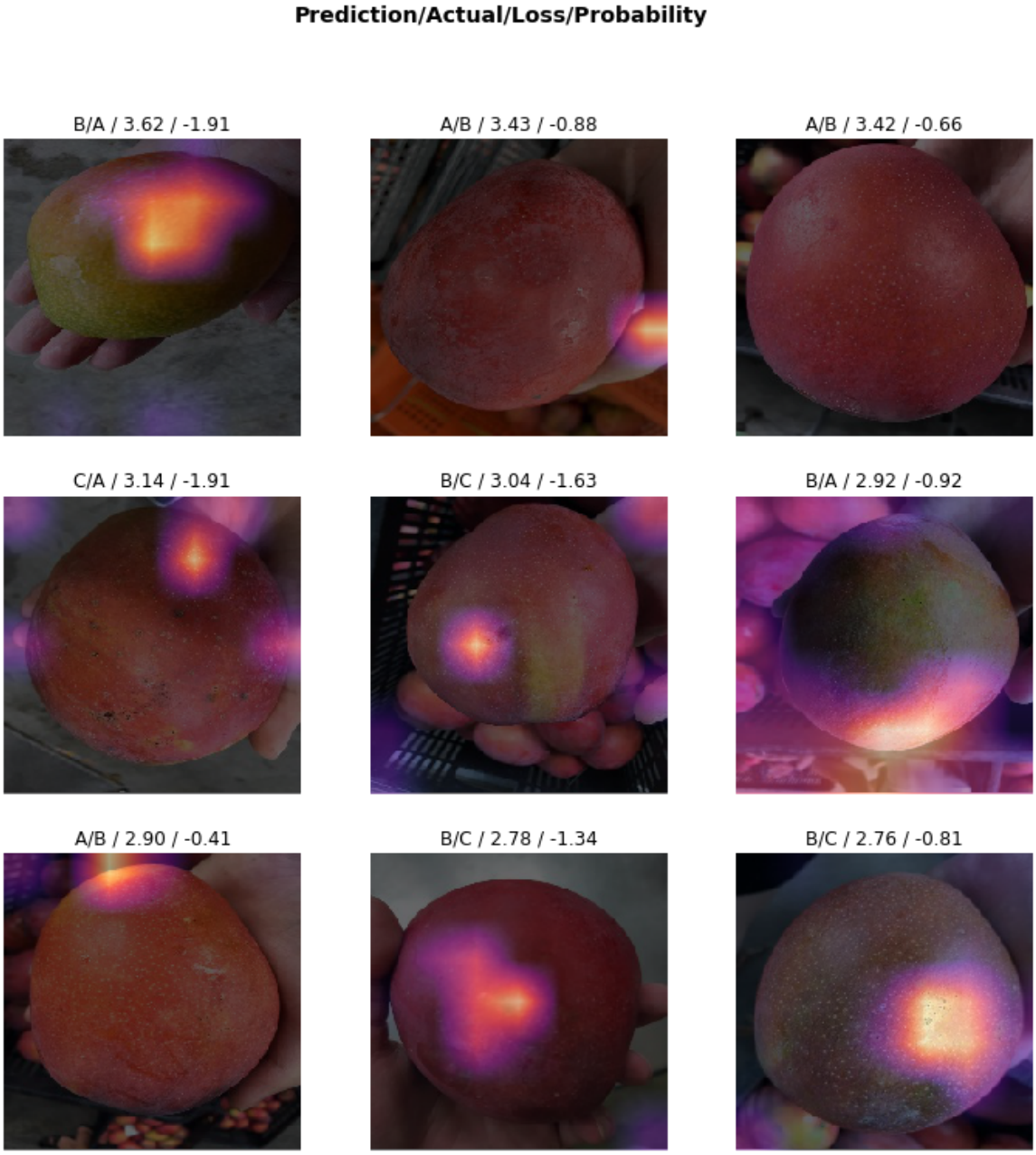
In [65]:

```
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```



In [75]:

```
interp.plot_top_losses(9,heatmap=True)
```



雙模型

first

In [76]:

```
path = os.getcwd()+"/train_valid_test"
data = ImageList.from_folder(path)
data=data.split_by_folder('train','valid')
func = lambda x: str(x)[-5] if (str(x)[-5])=='C' else 'O'
data=data.label_from_func(func)
tfms = get_transforms(do_flip=True,flip_vert=True)
bs=32
data=data.transform(tfms) \
          .databunch(bs=bs) \
          .normalize(imagenet_stats)
```

In [77]:

```
arch = models.densenet121
```

In [78]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True)
```

In [79]:

```
callbacks = [
SaveModelCallback(learn, monitor='recall',mode='max',name='densenet121_first')
,ShowGraph(learn),
]
learn.callbacks = callbacks
```

In [80]:

```
learn.summary()
```

Out[80]:

```
Sequential
==========================================================================
==
Layer (type)          Output Shape          Param #    Trainable
==========================================================================
==
Conv2d                [64, 112, 112]        9,408      False
--------------------------------------------------------------------------
--
BatchNorm2d           [64, 112, 112]        128        True
--------------------------------------------------------------------------
--
ReLU                  [64, 112, 112]        0          False
--------------------------------------------------------------------------
--
MaxPool2d             [64, 56, 56]          0          False
--------------------------------------------------------------------------
--
BatchNorm2d           [64, 56, 56]          128        True
--------------------------------------------------------------------------
--
ReLU                  [64, 56, 56]          0          False
--------------------------------------------------------------------------
--
Conv2d                [128, 56, 56]         8,192      False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
--
Conv2d                [32, 56, 56]          36,864     False
--------------------------------------------------------------------------
--
BatchNorm2d           [96, 56, 56]          192        True
--------------------------------------------------------------------------
--
ReLU                  [96, 56, 56]          0          False
--------------------------------------------------------------------------
--
Conv2d                [128, 56, 56]         12,288     False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
--
Conv2d                [32, 56, 56]          36,864     False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
```

Conv2d	[128, 56, 56]	16,384	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[160, 56, 56]	320	True
ReLU	[160, 56, 56]	0	False
Conv2d	[128, 56, 56]	20,480	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[192, 56, 56]	384	True
ReLU	[192, 56, 56]	0	False
Conv2d	[128, 56, 56]	24,576	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[224, 56, 56]	448	True
ReLU	[224, 56, 56]	0	False
Conv2d	[128, 56, 56]	28,672	False
BatchNorm2d	[128, 56, 56]	256	True

ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[256, 56, 56]	512	True
ReLU	[256, 56, 56]	0	False
Conv2d	[128, 56, 56]	32,768	False
AvgPool2d	[128, 28, 28]	0	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	16,384	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[160, 28, 28]	320	True
ReLU	[160, 28, 28]	0	False
Conv2d	[128, 28, 28]	20,480	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[192, 28, 28]	384	True
ReLU	[192, 28, 28]	0	False
Conv2d	[128, 28, 28]	24,576	False

BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[224, 28, 28]	448	True
ReLU	[224, 28, 28]	0	False
Conv2d	[128, 28, 28]	28,672	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[256, 28, 28]	512	True
ReLU	[256, 28, 28]	0	False
Conv2d	[128, 28, 28]	32,768	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[288, 28, 28]	576	True
ReLU	[288, 28, 28]	0	False
Conv2d	[128, 28, 28]	36,864	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False

Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[320, 28, 28]	640	True
ReLU	[320, 28, 28]	0	False
Conv2d	[128, 28, 28]	40,960	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[352, 28, 28]	704	True
ReLU	[352, 28, 28]	0	False
Conv2d	[128, 28, 28]	45,056	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[384, 28, 28]	768	True
ReLU	[384, 28, 28]	0	False
Conv2d	[128, 28, 28]	49,152	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[416, 28, 28]	832	True

ReLU	[416, 28, 28]	0	False
Conv2d	[128, 28, 28]	53,248	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[448, 28, 28]	896	True
ReLU	[448, 28, 28]	0	False
Conv2d	[128, 28, 28]	57,344	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[480, 28, 28]	960	True
ReLU	[480, 28, 28]	0	False
Conv2d	[128, 28, 28]	61,440	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[512, 28, 28]	1,024	True
ReLU	[512, 28, 28]	0	False
Conv2d	[256, 28, 28]	131,072	False
AvgPool2d	[256, 14, 14]	0	False

BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[128, 14, 14]	32,768	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[288, 14, 14]	576	True
ReLU	[288, 14, 14]	0	False
Conv2d	[128, 14, 14]	36,864	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[320, 14, 14]	640	True
ReLU	[320, 14, 14]	0	False
Conv2d	[128, 14, 14]	40,960	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[352, 14, 14]	704	True
ReLU	[352, 14, 14]	0	False

Conv2d	[128, 14, 14]	45,056	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[384, 14, 14]	768	True
ReLU	[384, 14, 14]	0	False
Conv2d	[128, 14, 14]	49,152	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[416, 14, 14]	832	True
ReLU	[416, 14, 14]	0	False
Conv2d	[128, 14, 14]	53,248	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[448, 14, 14]	896	True
ReLU	[448, 14, 14]	0	False
Conv2d	[128, 14, 14]	57,344	False
BatchNorm2d	[128, 14, 14]	256	True

ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[480, 14, 14]	960	True
ReLU	[480, 14, 14]	0	False
Conv2d	[128, 14, 14]	61,440	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[512, 14, 14]	1,024	True
ReLU	[512, 14, 14]	0	False
Conv2d	[128, 14, 14]	65,536	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[544, 14, 14]	1,088	True
ReLU	[544, 14, 14]	0	False
Conv2d	[128, 14, 14]	69,632	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[576, 14, 14]	1,152	True

ReLU	[576, 14, 14]	0	False
Conv2d	[128, 14, 14]	73,728	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[608, 14, 14]	1,216	True
ReLU	[608, 14, 14]	0	False
Conv2d	[128, 14, 14]	77,824	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[640, 14, 14]	1,280	True
ReLU	[640, 14, 14]	0	False
Conv2d	[128, 14, 14]	81,920	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[672, 14, 14]	1,344	True
ReLU	[672, 14, 14]	0	False
Conv2d	[128, 14, 14]	86,016	False

BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[704, 14, 14]	1,408	True
ReLU	[704, 14, 14]	0	False
Conv2d	[128, 14, 14]	90,112	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[736, 14, 14]	1,472	True
ReLU	[736, 14, 14]	0	False
Conv2d	[128, 14, 14]	94,208	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[768, 14, 14]	1,536	True
ReLU	[768, 14, 14]	0	False
Conv2d	[128, 14, 14]	98,304	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False

Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[800, 14, 14]	1,600	True
ReLU	[800, 14, 14]	0	False
Conv2d	[128, 14, 14]	102,400	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[832, 14, 14]	1,664	True
ReLU	[832, 14, 14]	0	False
Conv2d	[128, 14, 14]	106,496	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[864, 14, 14]	1,728	True
ReLU	[864, 14, 14]	0	False
Conv2d	[128, 14, 14]	110,592	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[896, 14, 14]	1,792	True
ReLU	[896, 14, 14]	0	False

Conv2d	[128, 14, 14]	114,688	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[928, 14, 14]	1,856	True
ReLU	[928, 14, 14]	0	False
Conv2d	[128, 14, 14]	118,784	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[960, 14, 14]	1,920	True
ReLU	[960, 14, 14]	0	False
Conv2d	[128, 14, 14]	122,880	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[992, 14, 14]	1,984	True
ReLU	[992, 14, 14]	0	False
Conv2d	[128, 14, 14]	126,976	False
BatchNorm2d	[128, 14, 14]	256	True

ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[1024, 14, 14]	2,048	True
ReLU	[1024, 14, 14]	0	False
Conv2d	[512, 14, 14]	524,288	False
AvgPool2d	[512, 7, 7]	0	False
BatchNorm2d	[512, 7, 7]	1,024	True
ReLU	[512, 7, 7]	0	False
Conv2d	[128, 7, 7]	65,536	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[544, 7, 7]	1,088	True
ReLU	[544, 7, 7]	0	False
Conv2d	[128, 7, 7]	69,632	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[576, 7, 7]	1,152	True
ReLU	[576, 7, 7]	0	False

Conv2d	[128, 7, 7]	73,728	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[608, 7, 7]	1,216	True
ReLU	[608, 7, 7]	0	False
Conv2d	[128, 7, 7]	77,824	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[640, 7, 7]	1,280	True
ReLU	[640, 7, 7]	0	False
Conv2d	[128, 7, 7]	81,920	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[672, 7, 7]	1,344	True
ReLU	[672, 7, 7]	0	False
Conv2d	[128, 7, 7]	86,016	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False

Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[704, 7, 7]	1,408	True
ReLU	[704, 7, 7]	0	False
Conv2d	[128, 7, 7]	90,112	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[736, 7, 7]	1,472	True
ReLU	[736, 7, 7]	0	False
Conv2d	[128, 7, 7]	94,208	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[768, 7, 7]	1,536	True
ReLU	[768, 7, 7]	0	False
Conv2d	[128, 7, 7]	98,304	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[800, 7, 7]	1,600	True

ReLU	[800, 7, 7]	0	False
Conv2d	[128, 7, 7]	102,400	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[832, 7, 7]	1,664	True
ReLU	[832, 7, 7]	0	False
Conv2d	[128, 7, 7]	106,496	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[864, 7, 7]	1,728	True
ReLU	[864, 7, 7]	0	False
Conv2d	[128, 7, 7]	110,592	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[896, 7, 7]	1,792	True
ReLU	[896, 7, 7]	0	False
Conv2d	[128, 7, 7]	114,688	False

BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[928, 7, 7]	1,856	True
ReLU	[928, 7, 7]	0	False
Conv2d	[128, 7, 7]	118,784	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[960, 7, 7]	1,920	True
ReLU	[960, 7, 7]	0	False
Conv2d	[128, 7, 7]	122,880	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[992, 7, 7]	1,984	True
ReLU	[992, 7, 7]	0	False
Conv2d	[128, 7, 7]	126,976	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False

```
____________________________________________________________________
__
BatchNorm2d          [1024, 7, 7]         2,048      True
____________________________________________________________________
__
AdaptiveAvgPool2d    [1024, 1, 1]         0          False
____________________________________________________________________
__
AdaptiveMaxPool2d    [1024, 1, 1]         0          False
____________________________________________________________________
__
Flatten              [2048]               0          False
____________________________________________________________________
__
BatchNorm1d          [2048]               4,096      True
____________________________________________________________________
__
Dropout              [2048]               0          False
____________________________________________________________________
__
Linear               [512]                1,049,088  True
____________________________________________________________________
__
ReLU                 [512]                0          False
____________________________________________________________________
__
BatchNorm1d          [512]                1,024      True
____________________________________________________________________
__
Dropout              [512]                0          False
____________________________________________________________________
__
Linear               [2]                  1,026      True
____________________________________________________________________
__

Total params: 8,009,090
Total trainable params: 1,138,882
Total non-trainable params: 6,870,208
Optimized with 'torch.optim.adam.Adam', betas=(0.9, 0.99)
Using true weight decay as discussed in https://www.fast.ai/2018/07/
02/adam-weight-decay/
Loss function : LabelSmoothingCrossEntropy
========================================================================
==
Callbacks functions applied
    SaveModelCallback
    ShowGraph
```

In [81]:

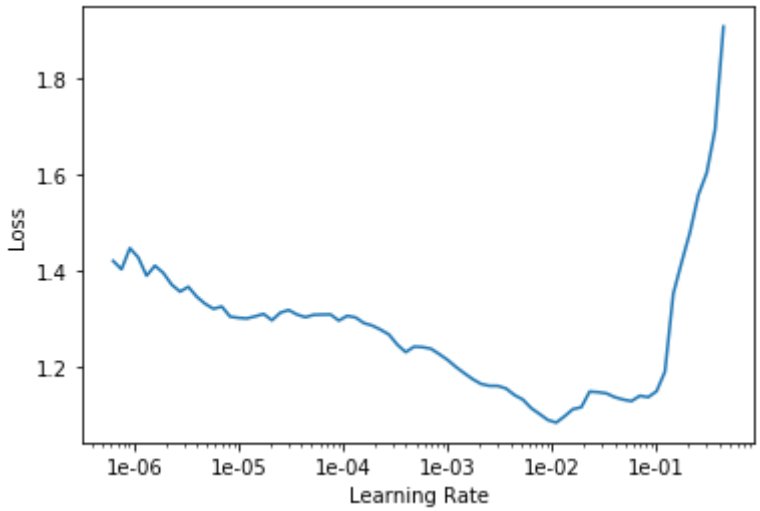
```
learn.lr_find()
learn.recorder.plot()
```

0.00% [0/1 00:00<00:00]

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time

55.35% [88/159 00:47<00:38 3.5211]

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

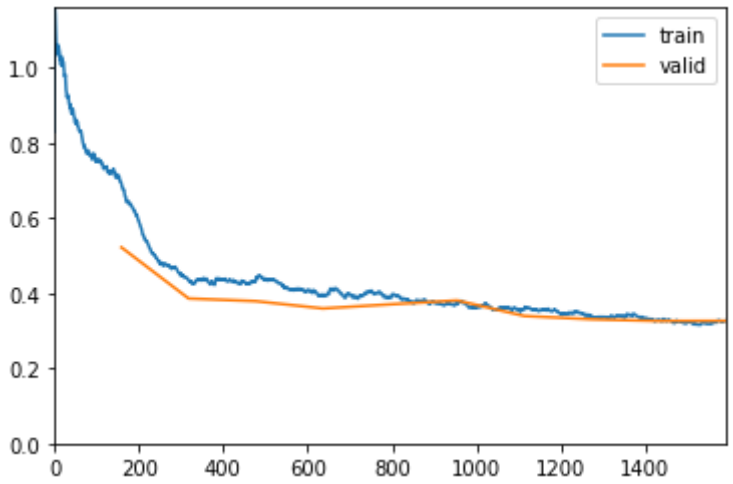


In [82]:

```
#first train
learn.fit_one_cycle(10,max_lr=7e-03)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	0.693996	0.522069	0.866511	0.866511	0.858445	01:35
1	0.437144	0.386024	0.886807	0.886807	0.839220	01:36
2	0.430094	0.378927	0.909446	0.909446	0.893324	01:36
3	0.395680	0.359441	0.909446	0.909446	0.892853	01:35
4	0.390542	0.370396	0.901639	0.901639	0.885651	01:35
5	0.372953	0.380753	0.899297	0.899297	0.872855	01:35
6	0.358965	0.339472	0.921155	0.921155	0.905328	01:36
7	0.338698	0.330683	0.915691	0.915691	0.894367	01:36
8	0.329461	0.326192	0.919594	0.919594	0.906753	01:35
9	0.326399	0.326472	0.921155	0.921155	0.912154	01:36

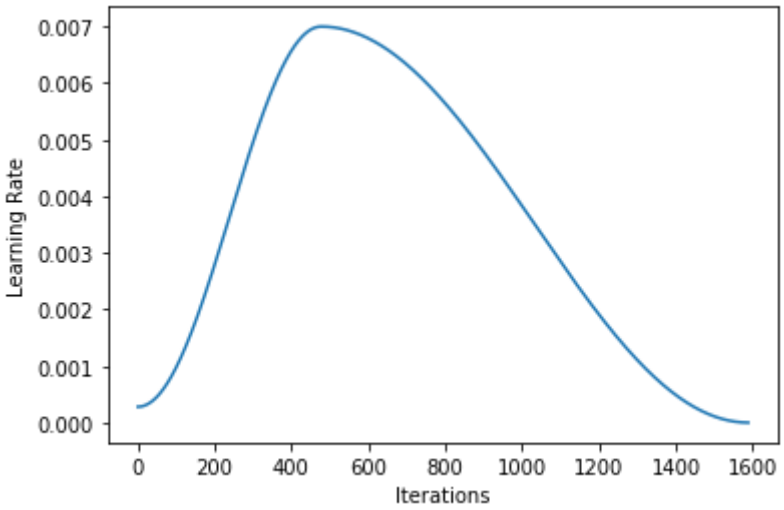
```
Better model found at epoch 0 with recall value: 0.8665105700492859.
```




```
Better model found at epoch 1 with recall value: 0.8868072032928467.
Better model found at epoch 2 with recall value: 0.9094457626342773.
Better model found at epoch 6 with recall value: 0.9211553335189819.
```

In [83]:

```
learn.recorder.plot_lr()
```

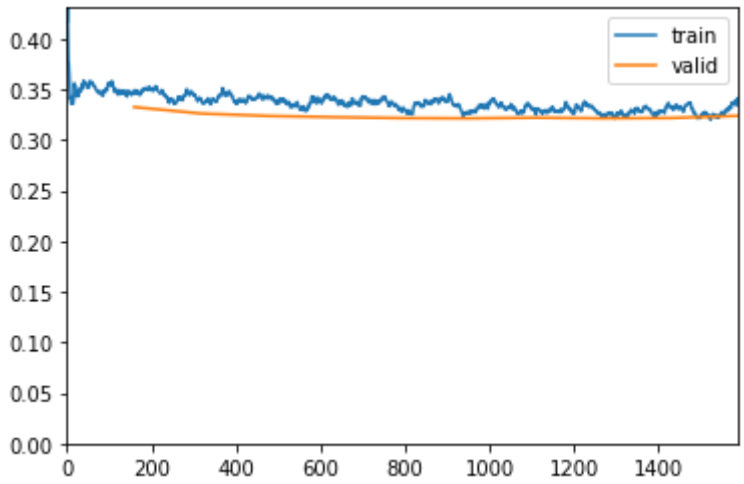


In [84]:

```
#fine tune
learn.fit_one_cycle(10,max_lr=7e-05)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	0.346818	0.332485	0.925839	0.925839	0.912643	01:35
1	0.339842	0.326113	0.925839	0.925839	0.914011	01:36
2	0.344240	0.323721	0.925059	0.925059	0.909648	01:36
3	0.340101	0.322526	0.927400	0.927400	0.917619	01:35
4	0.328565	0.321593	0.925839	0.925839	0.913101	01:36
5	0.325990	0.321362	0.926620	0.926620	0.915595	01:36
6	0.330546	0.322012	0.925059	0.925058	0.910115	01:36
7	0.325585	0.321370	0.925839	0.925839	0.913557	01:36
8	0.332860	0.321710	0.925839	0.925839	0.911722	01:36
9	0.341035	0.323853	0.920375	0.920375	0.913217	01:35

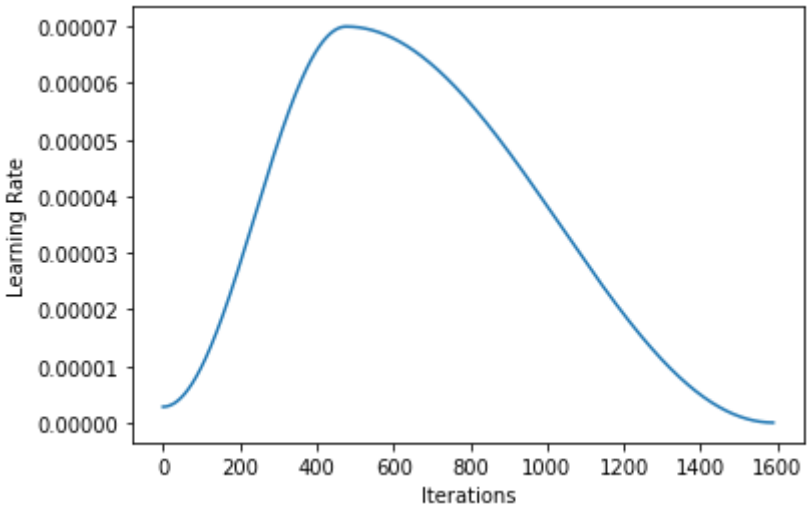
```
Better model found at epoch 0 with recall value: 0.9258391857147217.
```




```
Better model found at epoch 3 with recall value: 0.9274004697799683.
```

In [85]:

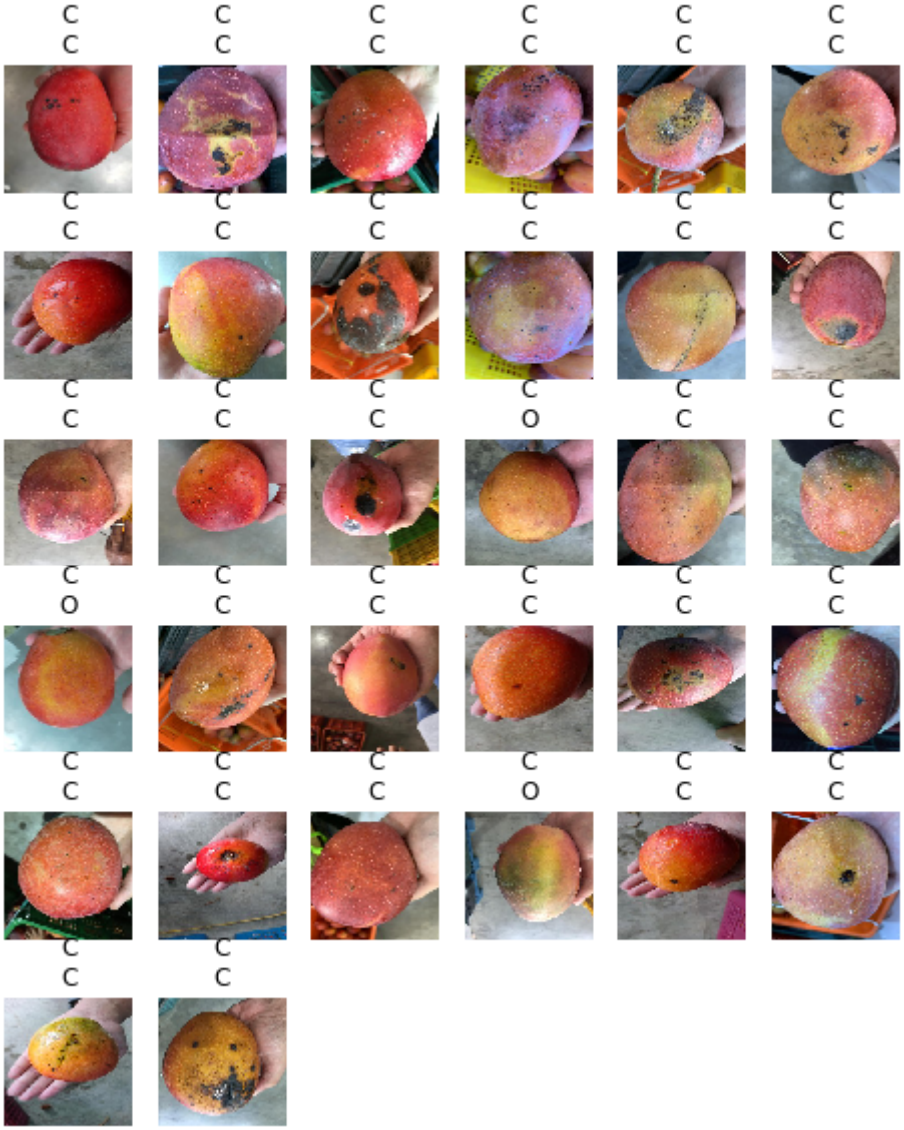
```
learn.recorder.plot_lr()
```



In [86]:

```
learn.show_results(ds_type=DatasetType.Valid, rows=6, figsize=(8,10))
```

Ground truth
Predictions

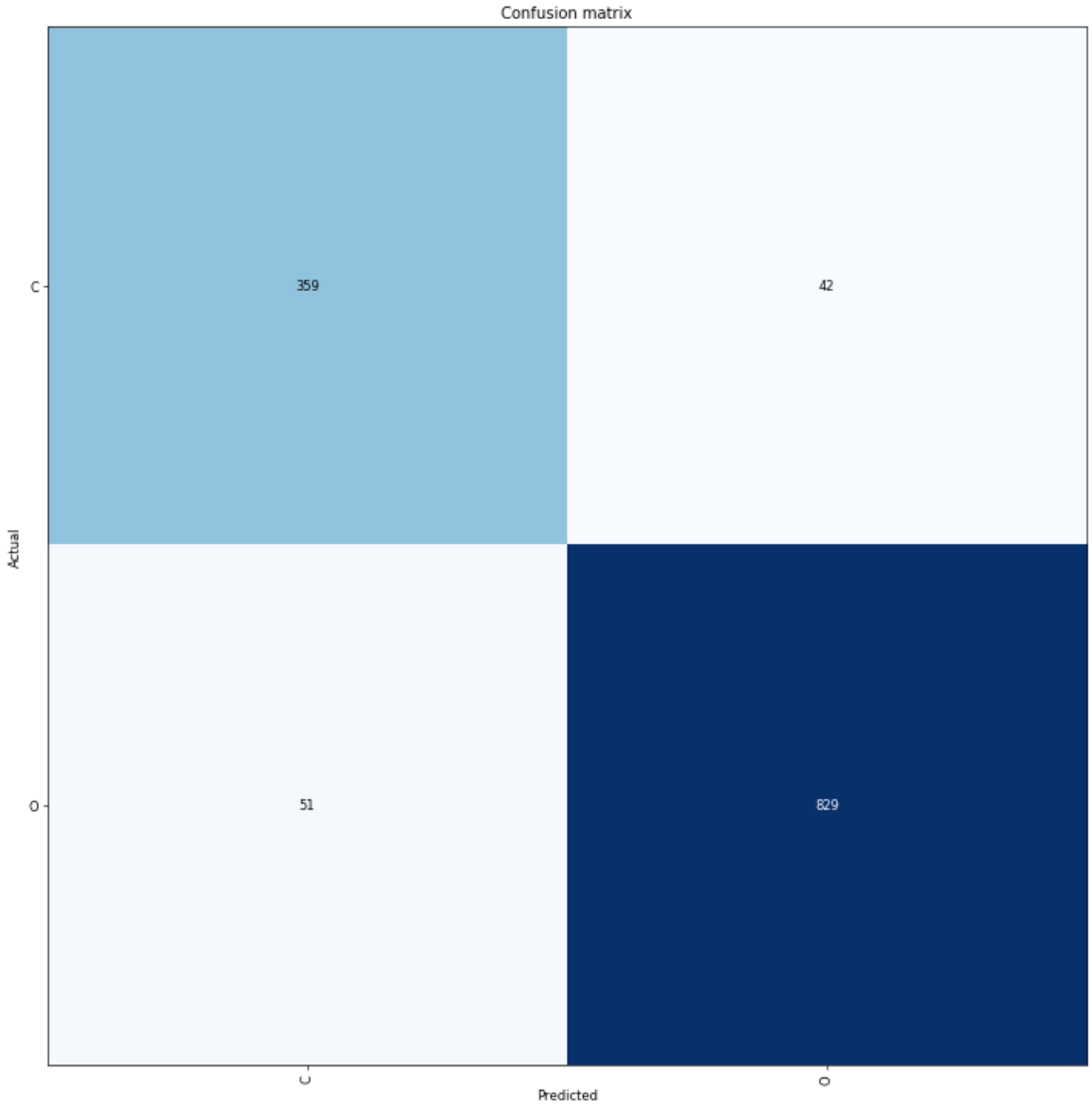


In [87]:

```
interp=ClassificationInterpretation.from_learner(learn)
```

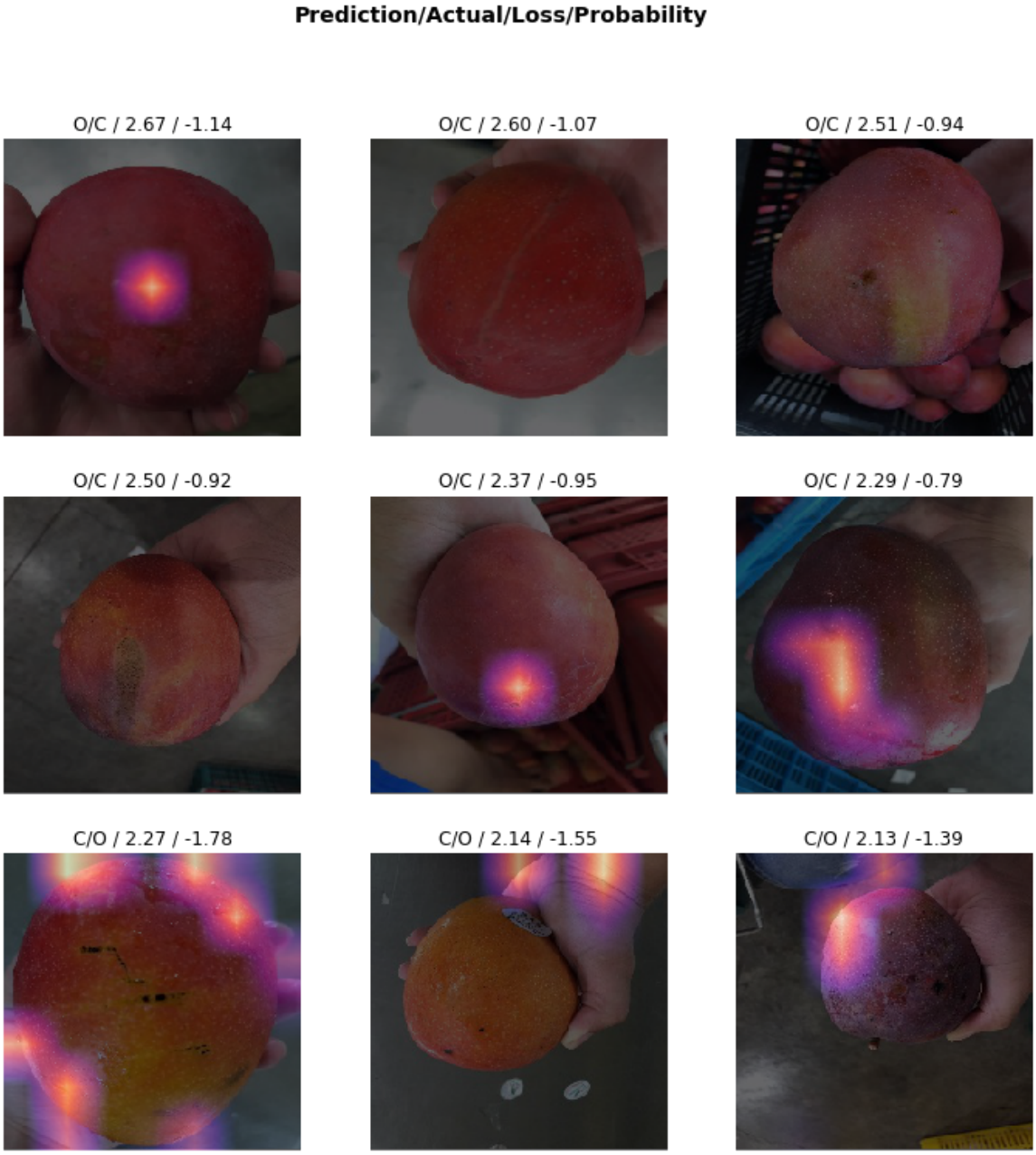
In [88]:

```
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```



In [89]:

```
interp.plot_top_losses(9,heatmap=True)
```



In [90]:

```
learn.save('doubledensenet121_one')
```

second

In [91]:

```
path = os.getcwd()+"/train_valid_testAB"
data = ImageList.from_folder(path)
data=data.split_by_folder('train','valid')
func = lambda x: str(x)[-5] if (str(x)[-5])=='A' else 'B'
data=data.label_from_func(func)
tfms = get_transforms(do_flip=True,flip_vert=True)
bs=32
data=data.transform(tfms) \
          .databunch(bs=bs) \
          .normalize(imagenet_stats)
```

In [92]:

```
arch = models.densenet121
```

In [93]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True)
```

In [94]:

```
callbacks = [
SaveModelCallback(learn, monitor='recall',mode='max',name='densenet121_second')
,ShowGraph(learn),
]
learn.callbacks = callbacks
```

In [95]:

```
learn.summary()
```

Out[95]:

```
Sequential
==========================================================================
==
Layer (type)          Output Shape          Param #    Trainable
==========================================================================
==
Conv2d                [64, 112, 112]        9,408      False
--------------------------------------------------------------------------
--
BatchNorm2d           [64, 112, 112]        128        True
--------------------------------------------------------------------------
--
ReLU                  [64, 112, 112]        0          False
--------------------------------------------------------------------------
--
MaxPool2d             [64, 56, 56]          0          False
--------------------------------------------------------------------------
--
BatchNorm2d           [64, 56, 56]          128        True
--------------------------------------------------------------------------
--
ReLU                  [64, 56, 56]          0          False
--------------------------------------------------------------------------
--
Conv2d                [128, 56, 56]         8,192      False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
--
Conv2d                [32, 56, 56]          36,864     False
--------------------------------------------------------------------------
--
BatchNorm2d           [96, 56, 56]          192        True
--------------------------------------------------------------------------
--
ReLU                  [96, 56, 56]          0          False
--------------------------------------------------------------------------
--
Conv2d                [128, 56, 56]         12,288     False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
--
Conv2d                [32, 56, 56]          36,864     False
--------------------------------------------------------------------------
--
BatchNorm2d           [128, 56, 56]         256        True
--------------------------------------------------------------------------
--
ReLU                  [128, 56, 56]         0          False
--------------------------------------------------------------------------
```

Conv2d	[128, 56, 56]	16,384	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[160, 56, 56]	320	True
ReLU	[160, 56, 56]	0	False
Conv2d	[128, 56, 56]	20,480	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[192, 56, 56]	384	True
ReLU	[192, 56, 56]	0	False
Conv2d	[128, 56, 56]	24,576	False
BatchNorm2d	[128, 56, 56]	256	True
ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[224, 56, 56]	448	True
ReLU	[224, 56, 56]	0	False
Conv2d	[128, 56, 56]	28,672	False
BatchNorm2d	[128, 56, 56]	256	True

ReLU	[128, 56, 56]	0	False
Conv2d	[32, 56, 56]	36,864	False
BatchNorm2d	[256, 56, 56]	512	True
ReLU	[256, 56, 56]	0	False
Conv2d	[128, 56, 56]	32,768	False
AvgPool2d	[128, 28, 28]	0	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[128, 28, 28]	16,384	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[160, 28, 28]	320	True
ReLU	[160, 28, 28]	0	False
Conv2d	[128, 28, 28]	20,480	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[192, 28, 28]	384	True
ReLU	[192, 28, 28]	0	False
Conv2d	[128, 28, 28]	24,576	False

BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[224, 28, 28]	448	True
ReLU	[224, 28, 28]	0	False
Conv2d	[128, 28, 28]	28,672	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[256, 28, 28]	512	True
ReLU	[256, 28, 28]	0	False
Conv2d	[128, 28, 28]	32,768	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[288, 28, 28]	576	True
ReLU	[288, 28, 28]	0	False
Conv2d	[128, 28, 28]	36,864	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False

Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[320, 28, 28]	640	True
ReLU	[320, 28, 28]	0	False
Conv2d	[128, 28, 28]	40,960	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[352, 28, 28]	704	True
ReLU	[352, 28, 28]	0	False
Conv2d	[128, 28, 28]	45,056	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[384, 28, 28]	768	True
ReLU	[384, 28, 28]	0	False
Conv2d	[128, 28, 28]	49,152	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[416, 28, 28]	832	True

ReLU	[416, 28, 28]	0	False
Conv2d	[128, 28, 28]	53,248	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[448, 28, 28]	896	True
ReLU	[448, 28, 28]	0	False
Conv2d	[128, 28, 28]	57,344	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[480, 28, 28]	960	True
ReLU	[480, 28, 28]	0	False
Conv2d	[128, 28, 28]	61,440	False
BatchNorm2d	[128, 28, 28]	256	True
ReLU	[128, 28, 28]	0	False
Conv2d	[32, 28, 28]	36,864	False
BatchNorm2d	[512, 28, 28]	1,024	True
ReLU	[512, 28, 28]	0	False
Conv2d	[256, 28, 28]	131,072	False
AvgPool2d	[256, 14, 14]	0	False

BatchNorm2d	[256, 14, 14]	512	True
ReLU	[256, 14, 14]	0	False
Conv2d	[128, 14, 14]	32,768	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[288, 14, 14]	576	True
ReLU	[288, 14, 14]	0	False
Conv2d	[128, 14, 14]	36,864	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[320, 14, 14]	640	True
ReLU	[320, 14, 14]	0	False
Conv2d	[128, 14, 14]	40,960	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[352, 14, 14]	704	True
ReLU	[352, 14, 14]	0	False

Conv2d	[128, 14, 14]	45,056	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[384, 14, 14]	768	True
ReLU	[384, 14, 14]	0	False
Conv2d	[128, 14, 14]	49,152	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[416, 14, 14]	832	True
ReLU	[416, 14, 14]	0	False
Conv2d	[128, 14, 14]	53,248	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[448, 14, 14]	896	True
ReLU	[448, 14, 14]	0	False
Conv2d	[128, 14, 14]	57,344	False
BatchNorm2d	[128, 14, 14]	256	True

ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[480, 14, 14]	960	True
ReLU	[480, 14, 14]	0	False
Conv2d	[128, 14, 14]	61,440	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[512, 14, 14]	1,024	True
ReLU	[512, 14, 14]	0	False
Conv2d	[128, 14, 14]	65,536	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[544, 14, 14]	1,088	True
ReLU	[544, 14, 14]	0	False
Conv2d	[128, 14, 14]	69,632	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[576, 14, 14]	1,152	True

ReLU	[576, 14, 14]	0	False
Conv2d	[128, 14, 14]	73,728	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[608, 14, 14]	1,216	True
ReLU	[608, 14, 14]	0	False
Conv2d	[128, 14, 14]	77,824	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[640, 14, 14]	1,280	True
ReLU	[640, 14, 14]	0	False
Conv2d	[128, 14, 14]	81,920	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[672, 14, 14]	1,344	True
ReLU	[672, 14, 14]	0	False
Conv2d	[128, 14, 14]	86,016	False

BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[704, 14, 14]	1,408	True
ReLU	[704, 14, 14]	0	False
Conv2d	[128, 14, 14]	90,112	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[736, 14, 14]	1,472	True
ReLU	[736, 14, 14]	0	False
Conv2d	[128, 14, 14]	94,208	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[768, 14, 14]	1,536	True
ReLU	[768, 14, 14]	0	False
Conv2d	[128, 14, 14]	98,304	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False

Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[800, 14, 14]	1,600	True
ReLU	[800, 14, 14]	0	False
Conv2d	[128, 14, 14]	102,400	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[832, 14, 14]	1,664	True
ReLU	[832, 14, 14]	0	False
Conv2d	[128, 14, 14]	106,496	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[864, 14, 14]	1,728	True
ReLU	[864, 14, 14]	0	False
Conv2d	[128, 14, 14]	110,592	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[896, 14, 14]	1,792	True
ReLU	[896, 14, 14]	0	False

Conv2d	[128, 14, 14]	114,688	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[928, 14, 14]	1,856	True
ReLU	[928, 14, 14]	0	False
Conv2d	[128, 14, 14]	118,784	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[960, 14, 14]	1,920	True
ReLU	[960, 14, 14]	0	False
Conv2d	[128, 14, 14]	122,880	False
BatchNorm2d	[128, 14, 14]	256	True
ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[992, 14, 14]	1,984	True
ReLU	[992, 14, 14]	0	False
Conv2d	[128, 14, 14]	126,976	False
BatchNorm2d	[128, 14, 14]	256	True

ReLU	[128, 14, 14]	0	False
Conv2d	[32, 14, 14]	36,864	False
BatchNorm2d	[1024, 14, 14]	2,048	True
ReLU	[1024, 14, 14]	0	False
Conv2d	[512, 14, 14]	524,288	False
AvgPool2d	[512, 7, 7]	0	False
BatchNorm2d	[512, 7, 7]	1,024	True
ReLU	[512, 7, 7]	0	False
Conv2d	[128, 7, 7]	65,536	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[544, 7, 7]	1,088	True
ReLU	[544, 7, 7]	0	False
Conv2d	[128, 7, 7]	69,632	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[576, 7, 7]	1,152	True
ReLU	[576, 7, 7]	0	False

Conv2d	[128, 7, 7]	73,728	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[608, 7, 7]	1,216	True
ReLU	[608, 7, 7]	0	False
Conv2d	[128, 7, 7]	77,824	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[640, 7, 7]	1,280	True
ReLU	[640, 7, 7]	0	False
Conv2d	[128, 7, 7]	81,920	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[672, 7, 7]	1,344	True
ReLU	[672, 7, 7]	0	False
Conv2d	[128, 7, 7]	86,016	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False

Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[704, 7, 7]	1,408	True
ReLU	[704, 7, 7]	0	False
Conv2d	[128, 7, 7]	90,112	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[736, 7, 7]	1,472	True
ReLU	[736, 7, 7]	0	False
Conv2d	[128, 7, 7]	94,208	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[768, 7, 7]	1,536	True
ReLU	[768, 7, 7]	0	False
Conv2d	[128, 7, 7]	98,304	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[800, 7, 7]	1,600	True

ReLU	[800, 7, 7]	0	False
Conv2d	[128, 7, 7]	102,400	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[832, 7, 7]	1,664	True
ReLU	[832, 7, 7]	0	False
Conv2d	[128, 7, 7]	106,496	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[864, 7, 7]	1,728	True
ReLU	[864, 7, 7]	0	False
Conv2d	[128, 7, 7]	110,592	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[896, 7, 7]	1,792	True
ReLU	[896, 7, 7]	0	False
Conv2d	[128, 7, 7]	114,688	False

BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[928, 7, 7]	1,856	True
ReLU	[928, 7, 7]	0	False
Conv2d	[128, 7, 7]	118,784	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[960, 7, 7]	1,920	True
ReLU	[960, 7, 7]	0	False
Conv2d	[128, 7, 7]	122,880	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False
BatchNorm2d	[992, 7, 7]	1,984	True
ReLU	[992, 7, 7]	0	False
Conv2d	[128, 7, 7]	126,976	False
BatchNorm2d	[128, 7, 7]	256	True
ReLU	[128, 7, 7]	0	False
Conv2d	[32, 7, 7]	36,864	False

```
____________________________________________________________________
__
BatchNorm2d          [1024, 7, 7]         2,048      True
____________________________________________________________________
__
AdaptiveAvgPool2d    [1024, 1, 1]         0          False
____________________________________________________________________
__
AdaptiveMaxPool2d    [1024, 1, 1]         0          False
____________________________________________________________________
__
Flatten              [2048]               0          False
____________________________________________________________________
__
BatchNorm1d          [2048]               4,096      True
____________________________________________________________________
__
Dropout              [2048]               0          False
____________________________________________________________________
__
Linear               [512]                1,049,088  True
____________________________________________________________________
__
ReLU                 [512]                0          False
____________________________________________________________________
__
BatchNorm1d          [512]                1,024      True
____________________________________________________________________
__
Dropout              [512]                0          False
____________________________________________________________________
__
Linear               [2]                  1,026      True
____________________________________________________________________
__

Total params: 8,009,090
Total trainable params: 1,138,882
Total non-trainable params: 6,870,208
Optimized with 'torch.optim.adam.Adam', betas=(0.9, 0.99)
Using true weight decay as discussed in https://www.fast.ai/2018/07/
02/adam-weight-decay/
Loss function : LabelSmoothingCrossEntropy
========================================================================
==
Callbacks functions applied
    SaveModelCallback
    ShowGraph
```

In [96]:

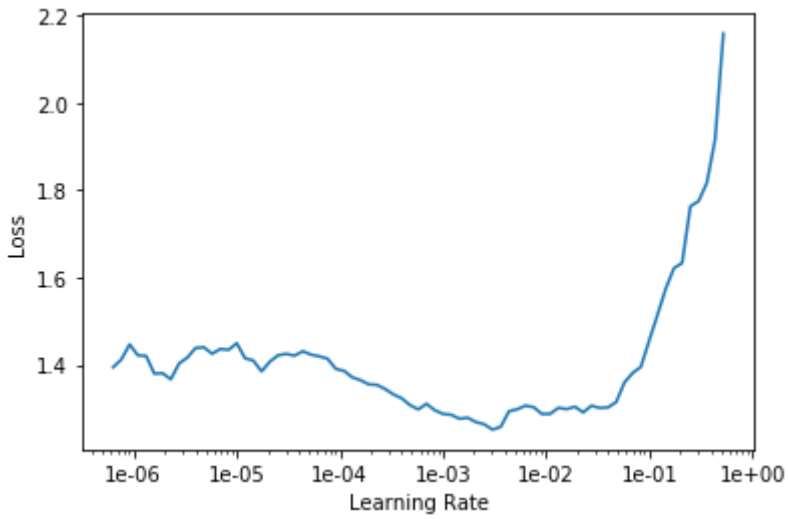
```
learn.lr_find()
learn.recorder.plot()
```

0.00% [0/1 00:00<00:00]

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time

81.65% [89/109 00:48<00:10 4.8183]

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

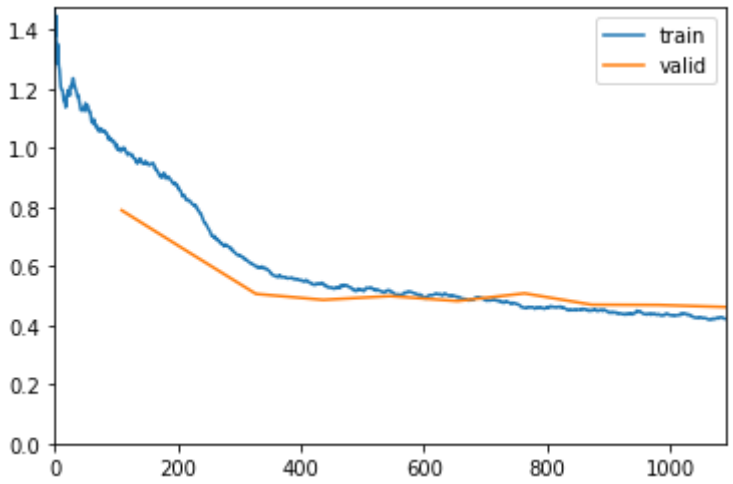


In [97]:

```
#first train
learn.fit_one_cycle(10,max_lr=3e-03)
```

epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	0.992789	0.788510	0.737500	0.737500	0.738662	01:08
1	0.824297	0.646635	0.751136	0.751136	0.745267	01:06
2	0.599633	0.505877	0.778409	0.778409	0.775917	01:06
3	0.541339	0.485746	0.803409	0.803409	0.803603	01:06
4	0.517158	0.498535	0.779545	0.779546	0.778026	01:06
5	0.495741	0.481757	0.813636	0.813636	0.811613	01:06
6	0.458741	0.507696	0.792045	0.792045	0.793885	01:06
7	0.448707	0.469328	0.820455	0.820455	0.821428	01:06
8	0.436236	0.468393	0.812500	0.812500	0.812216	01:06
9	0.421738	0.461413	0.805682	0.805682	0.805253	01:06

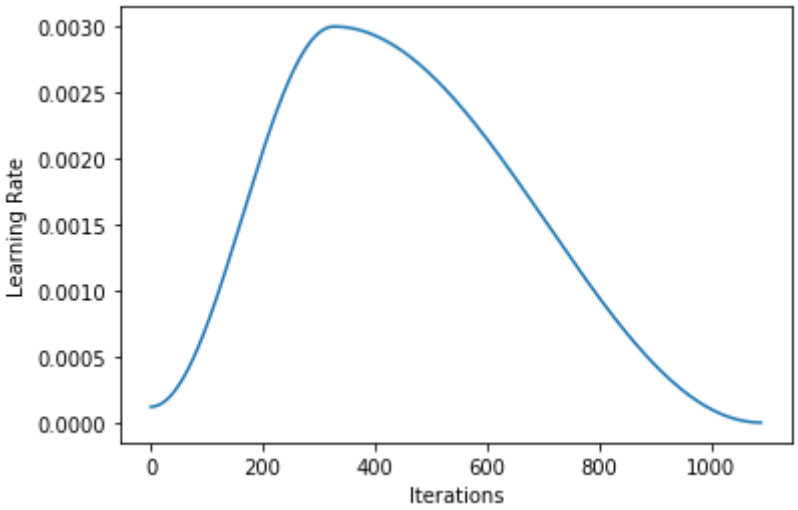
```
Better model found at epoch 0 with recall value: 0.737500011920929.
```




```
Better model found at epoch 1 with recall value: 0.7511364221572876.
Better model found at epoch 2 with recall value: 0.7784091234207153.
Better model found at epoch 3 with recall value: 0.8034090995788574.
Better model found at epoch 5 with recall value: 0.8136364221572876.
Better model found at epoch 7 with recall value: 0.8204545974731445.
```

In [98]:

```
learn.recorder.plot_lr()
```

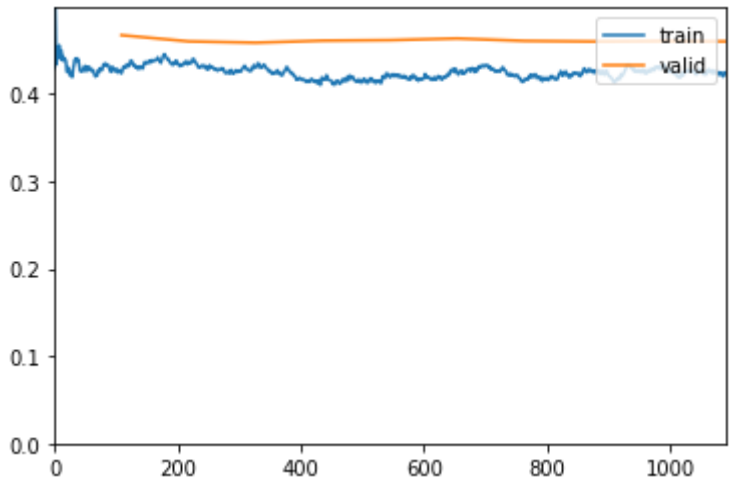


In [99]:

```
#fine tune
learn.fit_one_cycle(10,max_lr=3e-05)
```

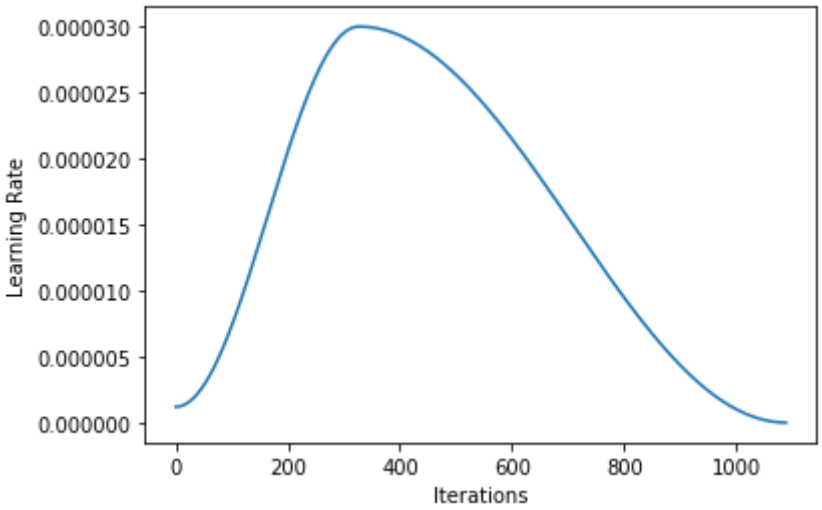
epoch	train_loss	valid_loss	accuracy	recall	f_beta	time
0	0.424905	0.467233	0.822727	0.822727	0.823708	01:06
1	0.436636	0.460131	0.820455	0.820455	0.819927	01:06
2	0.430744	0.458609	0.821591	0.821591	0.821955	01:06
3	0.418571	0.460963	0.818182	0.818182	0.818755	01:06
4	0.419781	0.461553	0.819318	0.819318	0.820100	01:06
5	0.426856	0.463443	0.821591	0.821591	0.822800	01:06
6	0.414350	0.460678	0.818182	0.818182	0.818578	01:06
7	0.427119	0.460009	0.818182	0.818182	0.818190	01:06
8	0.429805	0.460445	0.820455	0.820455	0.820949	01:06
9	0.424452	0.460113	0.818182	0.818182	0.817642	01:06

```
Better model found at epoch 0 with recall value: 0.8227273225784302.
```



In [100]:

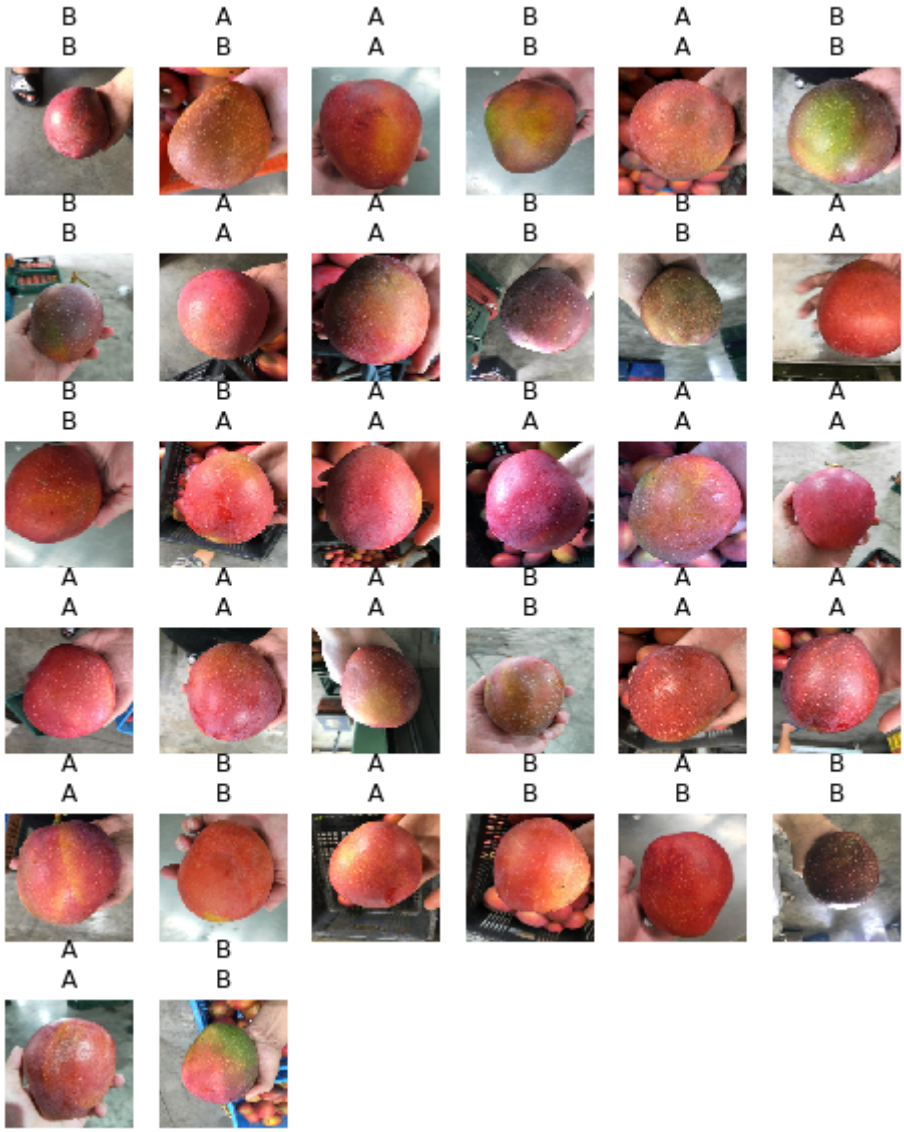
```
learn.recorder.plot_lr()
```



In [101]:

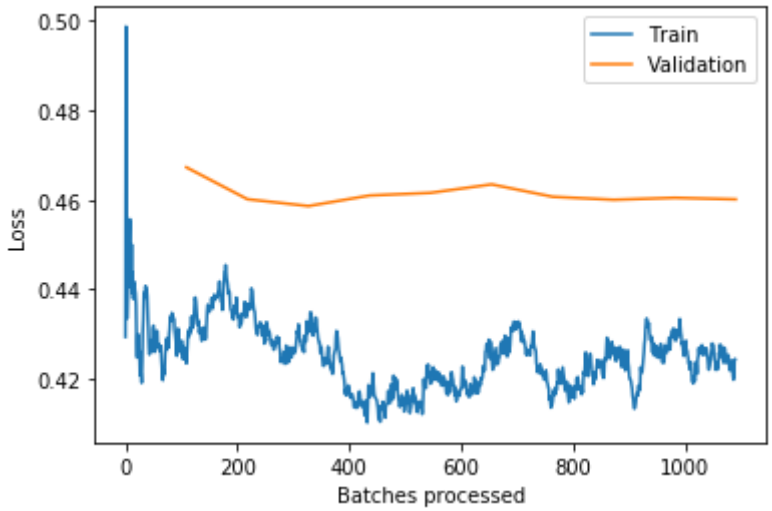
```
learn.show_results(ds_type=DatasetType.Valid, rows=6, figsize=(8,10))
```

Ground truth
Predictions



In [102]:

```
learn.recorder.plot_losses()
```

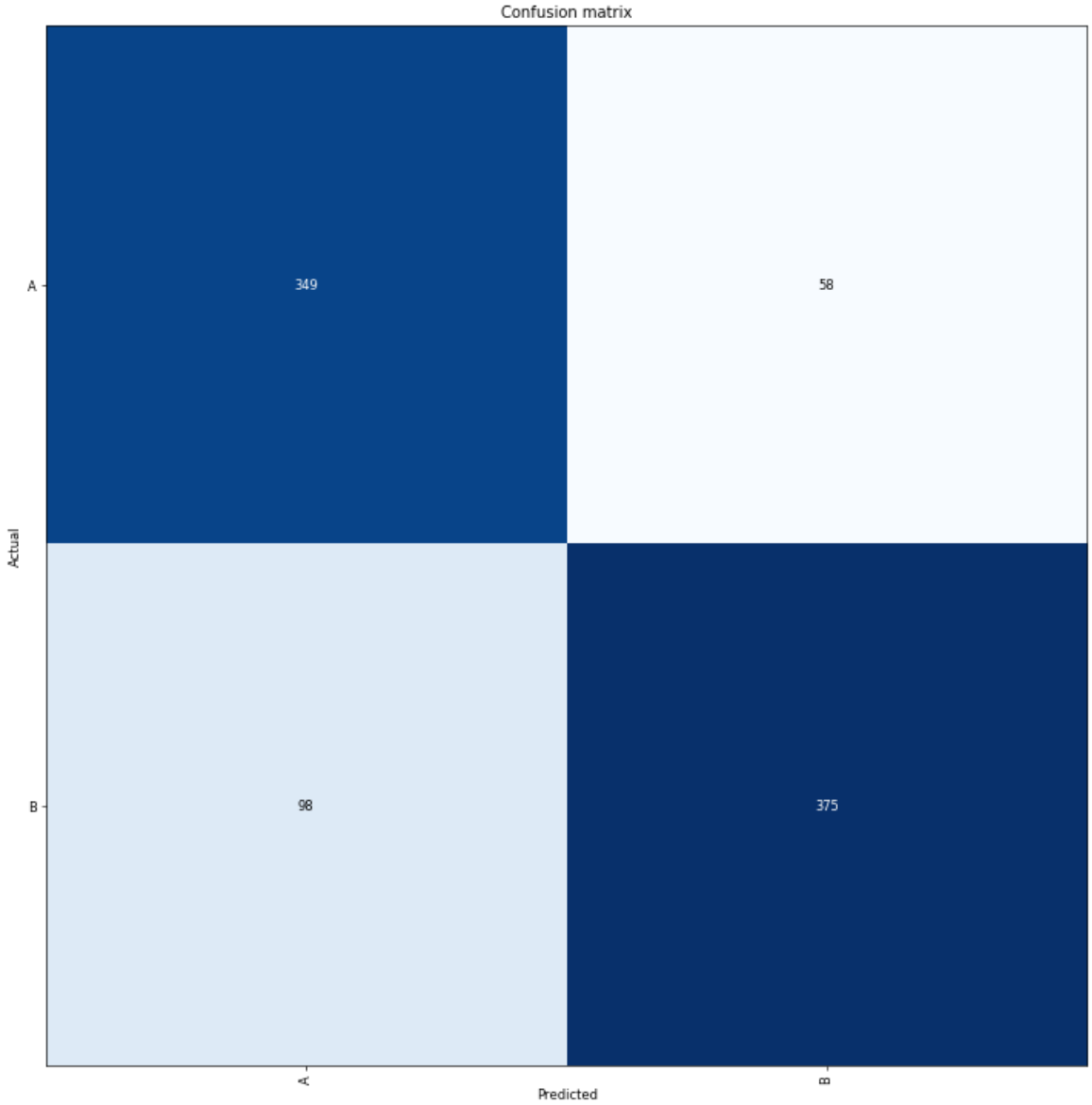


In [103]:

```
interp=ClassificationInterpretation.from_learner(learn)
```

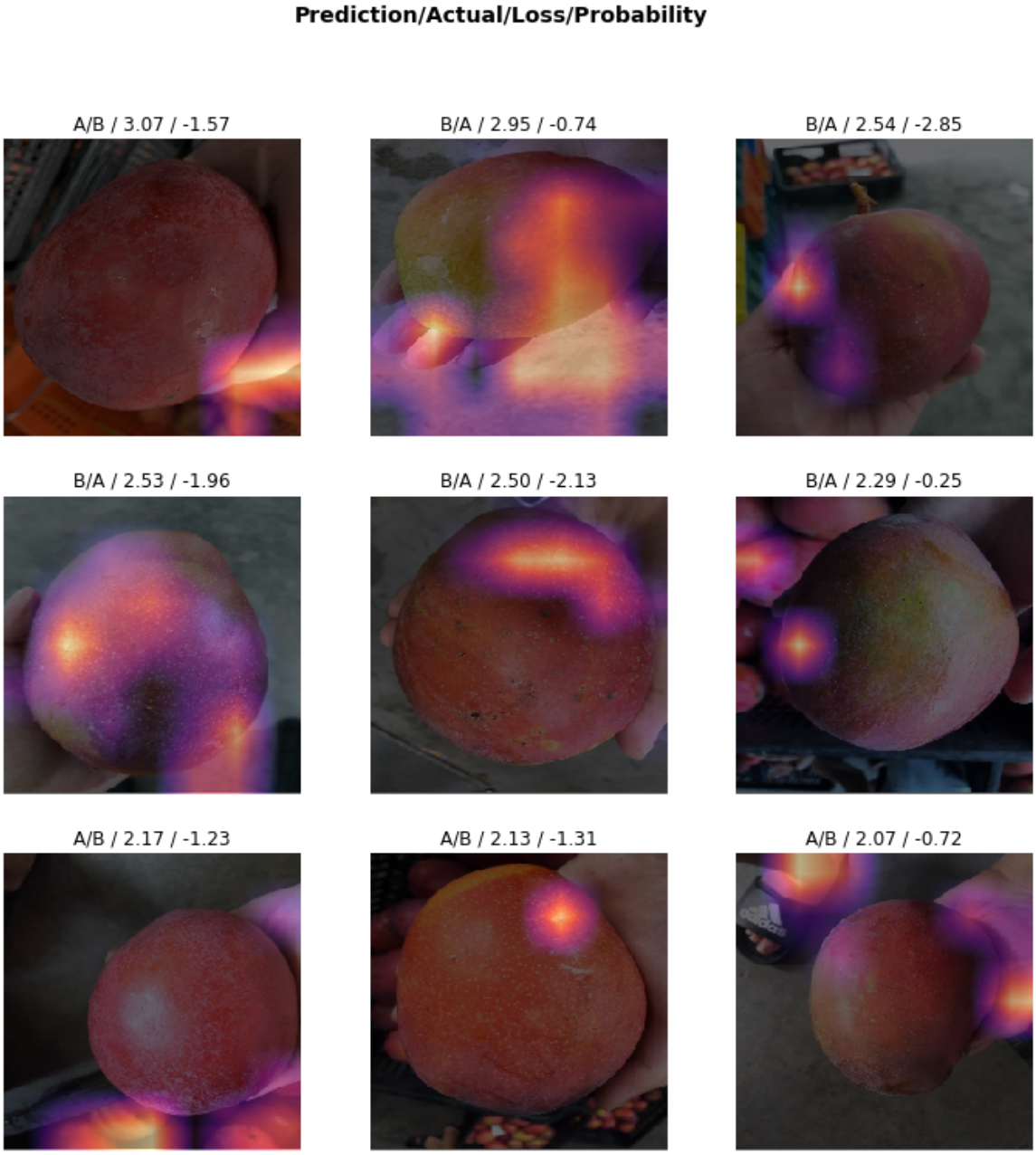
In [104]:

```
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```


Confusion matrix
A
B
349
58
98
375
Actual
A
B
Predicted

In [105]:

```
interp.plot_top_losses(9,heatmap=True)
```



In [106]:

```
learn.save('doubledensenet121_two')
```

預測test data、製作csv

單模型

In [ ]:

```
arch=models.resnet34
```

In [ ]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True).load('resnet34_10_epo
chs')
```

In [ ]:

```
import csv
with open('resnet34_10_epochs.csv', 'w', newline='') as csvfile:
     path=os.getcwd()+"/train_valid_test/test"
     writer = csv.writer(csvfile, delimiter=' ')
     writer.writerow(['image_id', 'label'])
     for i in sorted(os.listdir(path)):
            cat, tensor, probs = learn.predict(open_image(path+"/"+i))
            writer.writerow([i, cat.obj])
```

In [ ]:

```
arch=models.densenet121
```

In [ ]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True).load('densenet121_10_
epochs')
```

In [ ]:

```
import csv
with open('densenet121_10_epochs.csv', 'w', newline='') as csvfile:
     path=os.getcwd()+"/train_valid_test/test"
     writer = csv.writer(csvfile, delimiter=' ')
     writer.writerow(['image_id', 'label'])
     for i in sorted(os.listdir(path)):
            cat, tensor, probs = learn.predict(open_image(path+"/"+i))
            writer.writerow([i, cat.obj])
```

雙模型

In [ ]:

```
arch=models.densenet121
```

In [ ]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True).load('doubledensenet1
21_one')
```

In [ ]:

```
with open('doubledensenet121.csv', 'w', newline='') as csvfile:
     path=os.getcwd()+"/train_valid_test/test"
     writer = csv.writer(csvfile, delimiter=' ')
     writer.writerow(['image_id', 'label'])
     for i in sorted(os.listdir(path)):
            cat, tensor, probs = learn.predict(open_image(path+"/"+i))
            if cat.obj!='C':
                shutil.copyfile(path+"/"+i,os.getcwd()+'/Test_AB'+"/"+i)
            else:
                writer.writerow([i, cat.obj])
```

In [ ]:

```
learn = cnn_learner(data,arch,metrics=[accuracy,Recall('weighted'),FBeta('macro'
)],loss_func=LabelSmoothingCrossEntropy(),pretrained=True).load('doubledensenet1
21_two')
```

In [ ]:

```
with open('doubledensenet121.csv', 'a', newline='') as csvfile:
     path=os.getcwd()+'/Test_AB'
     writer = csv.writer(csvfile, delimiter=' ')
     for i in sorted(os.listdir(path)):
            cat, tensor, probs = learn.predict(open_image(path+"/"+i))
            if cat.obj == "A":
                writer.writerow([i, cat.obj])
            else:
                writer.writerow([i, 'B'])
```