In [1]:
```
import pandas as pd
import os
import numpy as np
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.model_selection import StratifiedKFold
from fastai import *
from fastai.vision import *
from fastai.callbacks import *
```

In [2]:
```
df_Test_Train = pd.read_csv('Fastai_dataset_usable.csv')
df_Test_Train.drop(df_Test_Train.columns[df_Test_Train.columns.str.contains('unnamed',case = False)],axis = 1, inplace = True)

df_Test_Train.head(100)
```

Out[2]:

	Crp_Filepath	clutch	Day	egg_number	sex
0	Cropped_Egg_images/Clutch1_D18/egg2.JPG	1	18	2	Female
1	Cropped_Egg_images/Clutch1_D18/egg3.JPG	1	18	3	Female
2	Cropped_Egg_images/Clutch1_D18/egg4.JPG	1	18	4	Male
3	Cropped_Egg_images/Clutch1_D18/egg6.JPG	1	18	6	Female
4	Cropped_Egg_images/Clutch1_D18/egg9.JPG	1	18	9	Female
5	Cropped_Egg_images/Clutch1_D18/egg11.JPG	1	18	11	Female
6	Cropped_Egg_images/Clutch1_D18/egg15.JPG	1	18	15	Female
7	Cropped_Egg_images/Clutch1_D18/egg16.JPG	1	18	16	Male
8	Cropped_Egg_images/Clutch1_D18/egg17.JPG	1	18	17	Female
9	Cropped_Egg_images/Clutch1_D18/egg19.JPG	1	18	19	Male
10	Cropped_Egg_images/Clutch1_D18/egg20.JPG	1	18	20	Male
11	Cropped_Egg_images/Clutch1_D18/egg21.JPG	1	18	21	Female

In [3]:
```
from sklearn.metrics import roc_auc_score

def auroc_score(input, target):
    input, target = input.cpu().numpy()[:,1], target.cpu().numpy()
    return roc_auc_score(target, input)

class AUROC(Callback):
    _order = -20 #Needs to run before the recorder

    def __init__(self, learn,extra=None, **kwargs): self.learn = learn
    def on_train_begin(self, **kwargs): self.learn.recorder.add_metric_names(['AUROC'])
    def on_epoch_begin(self, **kwargs): self.output, self.target = [], []

    def on_batch_end(self, last_target, last_output, train, **kwargs):
        if not train:
            self.output.append(last_output)
            self.target.append(last_target)

    def on_epoch_end(self, last_metrics, **kwargs):
        if len(self.output) > 0:
            output = torch.cat(self.output)
            target = torch.cat(self.target)
            preds = F.softmax(output, dim=1)
            metric = auroc_score(preds, target)
            return add_metrics(last_metrics, [metric])
```

Continued testing on previous best results

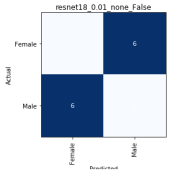
previous tests (initial_testing.ipynb)

Repeated Stratified K-Fold tests

- Resnet18
- No Transforms
- wd = 0.01
- 9 epochs
- normalization off
- max lr not set

In [8]:
```
'''
Results from this test found that 42.8% of the 35 models had an AUROC >=.5
The Std of the AUROCS was .142529 with a mean AUROC of .503
'''

# Parameters to very
epoch_cycles=9
np.random.seed(42)

# Tests to Perform
tests = [[models.resnet18 , .01, None, False]]

# Creating Frame Work
DFBig = pd.DataFrame(columns = ['test_name','model_arch', 'transforms', 'normalized',
                                'weight_decay', 'split_num', 'train_df'])
# Creating Stratified K folds
rskf = RepeatedStratifiedKFold(n_splits=5, n_repeats=6)

# For loop for test

for x in tests:
    wd = x[1]
    norm = x[3]
    arch = x[0]
    archstr = str(arch).split(' ')[1]
    if x[2] is not None:
        tsfmstr = 'modified'
    else:
        tsfmstr = 'none'
    test_name = archstr + '_' + str(wd) + '_' + tsfmstr + '_' + str(x[3])
    split_num = 1
    for train_index, val_index in rskf.split(df_Test_Train.index, df_Test_Train.sex):
        print(test_name)
        if norm:
            data_fold = (ImageList.from_df(df_Test_Train,'/home/jplineb/Chicken_Proj')
                         .split_by_idxs(train_index, val_index)
                         .label_from_df(cols='sex')
                         .transform(x[2], size=224)
                         .databunch(bs = 2)).normalize()
        else:
            data_fold = (ImageList.from_df(df_Test_Train,'/home/jplineb/Chicken_Proj')
                         .split_by_idxs(train_index, val_index)
                         .label_from_df(cols='sex')
                         .transform(x[2], size=224)
                         .databunch(bs = 2))

        learn = cnn_learner(data_fold, arch, metrics=error_rate, pretrained=True,callback_fns = [CSVLogger,AUROC, partial(EarlyStoppingCallback, monitor='AUROC', mode='max', min_delta=0.01, patience=100)], wd=wd)
        learn.fit_one_cycle(epoch_cycles)

        df_history = pd.read_csv('history.csv') # Appends to dataframe
        lowesterror = df_history.error_rate.idxmin()
        DFBig = DFBig.append({'test_name': test_name,
                              'model_arch': archstr,
                              'transforms': tsfmstr,
                              'normalized': str(norm),
                              'weight_decay': wd,
                              'split_num': split_num, # indicates kfold split
                              'error_rate_8' : df_history.error_rate[7],
                              'error_rate_9' : df_history.error_rate[8],
                              'AUROC_8': df_history.AUROC[7],
                              'AUROC_9': df_history.AUROC[8],
                              'train_df': df_history}, ignore_index = True)

        interp = ClassificationInterpretation.from_learner(learn)
        interp.plot_confusion_matrix(return_fig=True, title=test_name)
        split_num += 1
        DFBig.to_csv('DF_resnet18_01_epoch9.csv')
```

```
/usr/local/lib/python3.6/dist-packages/matplotlib/pyplot.py:514: RuntimeWarning: More than 20 figures have been opened. Figures created through the pyplot interface (`matplotlib.pyplot.figure`) are retained until explicitly closed and may consume too much memory. (To control this warning, see the rcParam `figure.max_open_warning`).
  max_open_warning, RuntimeWarning)
```



Results from the test:

df - DataFrame

Index	Unnamed: 0	test_name	model_arch	transforms	normalized	weight_decay	split_num	train_df	AUROC_8	AUROC_9	error_rate_8	error_rate_9
4	4	resnet1...	resnet18	none	False	1	5	epoc...	0.802469	0.777778	0.333333	0.277778
29	29	resnet1...	resnet18	none	False	1	30	epoc...	0.703704	0.777778	0.388889	0.333333
34	34	resnet1...	resnet18	none	False	1	35	epoc...	0.691358	0.728395	0.388889	0.277778
2	2	resnet1...	resnet18	none	False	1	3	epoc...	0.691358	0.691358	0.333333	0.333333
7	7	resnet1...	resnet18	none	False	1	8	epoc...	0.728395	0.691358	0.333333	0.388889
0	0	resnet1...	resnet18	none	False	1	1	epoc...	0.69	0.67	0.35	0.35
22	22	resnet1...	resnet18	none	False	1	23	epoc...	0.654321	0.654321	0.444444	0.333333
12	12	resnet1...	resnet18	none	False	1	13	epoc...	0.62963	0.62963	0.333333	0.277778
17	17	resnet1...	resnet18	none	False	1	18	epoc...	0.666667	0.62963	0.333333	0.388889
23	23	resnet1...	resnet18	none	False	1	24	epoc...	0.54321	0.567901	0.5	0.555556
27	27	resnet1...	resnet18	none	False	1	28	epoc...	0.604938	0.567901	0.388889	0.388889
1	1	resnet1...	resnet18	none	False	1	2	epoc...	0.48	0.56	0.6	0.55
8	8	resnet1...	resnet18	none	False	1	9	epoc...	0.641975	0.54321	0.5	0.555556
25	25	resnet1...	resnet18	none	False	1	26	epoc...	0.51	0.52	0.5	0.35
32	32	resnet1...	resnet18	none	False	1	33	epoc...	0.555556	0.518519	0.444444	0.5
19	19	resnet1...	resnet18	none	False	1	20	epoc...	0.432099	0.493827	0.5	0.5
3	3	resnet1...	resnet18	none	False	1	4	epoc...	0.419753	0.481481	0.5	0.444444
18	18	resnet1...	resnet18	none	False	1	19	epoc...	0.530864	0.481481	0.555556	0.611111
6	6	resnet1...	resnet18	none	False	1	7	epoc...	0.4	0.48	0.6	0.55
15	15	resnet1...	resnet18	none	False	1	16	epoc...	0.56	0.48	0.6	0.45

Format | Resize | Background color | Column min/max | Save and Close | Close

max_lr Investigation

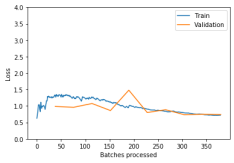
In [59]:
```
learn = cnn_learner(data, models.resnet18, metrics=error_rate, wd=0.5, pretrained=True,
                    callback_fns=[AUROC, partial(EarlyStoppingCallback, monitor='AUROC', mode='max', min_delta=0.01, patience=100)])
```

In [69]:
```
learn = cnn_learner(data, models.resnet18, metrics=error_rate, wd=0.5, pretrained=True,
                    callback_fns=[AUROC, partial(EarlyStoppingCallback, monitor='AUROC', mode='max', min_delta=0.01, patience=100)])
learn.fit_one_cycle(10, max_lr=0.03)
learn.recorder.plot_losses()
plt.ylim(0,4)
```

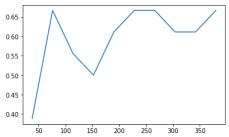
Total time: 00:23

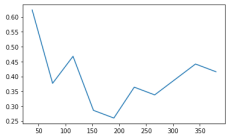
epoch	train_loss	valid_loss	error_rate	AUROC	time
0	1.333933	0.985290	0.388889	0.623377	00:02
1	1.283098	0.960671	0.666667	0.376623	00:02
2	1.272604	1.074930	0.555556	0.467532	00:02
3	1.131919	0.860839	0.500000	0.285714	00:02
4	0.947059	1.480110	0.611111	0.259740	00:02
5	0.909261	0.801205	0.666667	0.363636	00:02
6	0.835265	0.886731	0.666667	0.337662	00:02
7	0.795152	0.735863	0.611111	0.389610	00:02
8	0.744350	0.746402	0.611111	0.441558	00:02
9	0.716197	0.740237	0.666667	0.415584	00:02

Out[69]: (0, 4)



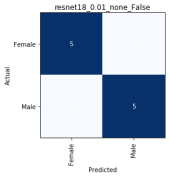
In [70]:
```
learn.recorder.plot_metrics()
```





In [28]:
```
interp.plot_top_losses(12, figsize=(15,15))
interp.plot_confusion_matrix(return_fig=True, title=test_name)
```

Out[28]:



New hyperparameter search

Transforms:

- resize (size=224)
- rotate (random +/- 15 deg)
- zoom (random scale from 0.85 to 1.15)
- flip (p=0.5)

Normalization: true

Search parameters:

In [83]:
```
max_lr = [0.01, 0.03, 0.1, 0.3]
wd = [0.01, 0.03, 0.1, 0.3, 1]
epoch = [5,7,9]
k =5
secs_per_epoch=2
```

In [84]:
```
# estimated time
np.sum([secs_per_epoch*e*len(max_lr)*len(wd) for e in epoch]) *k / 60
```

Out[84]: 70.0

In [4]:
```
'''From this test we found that:

* Resnet18
* 7 Epochs
* learning rate of 0.1
* weight decay of 1

and

* Resnet18
* 7 Epochs
* weight decay of 1
* learning rate of 0.3

returned the best results
'''

max_lr = [0.01, 0.03, 0.1, 0.3]
weight_decay = [0.01, 0.03, 0.1, 0.3, 1]
epoch_cycles = [5,7,9]
k=5
secs_per_epoch=2

transforms=[[RandTransform(tfm=TfmPixel (flip_lr), kwargs={}, p=0.5, resolved={}, do_run=True, is_random=True),
           RandTransform(tfm=TfmAffine (rotate), kwargs={'degrees': (-15.0, 15.0)}, p=0.75, resolved={}, do_run=True, is_random=True),
           RandTransform(tfm=TfmAffine (zoom), kwargs={'scale': (0.85, 1.15), 'row_pct': (0, 1), 'col_pct': (0, 1)}, p=0.5, resolved={}, do_run=True, is_random=True)],
            []]

# Creating Frame Work
DFBig = pd.DataFrame(columns = ['test_name','model_arch', 'transforms', 'normalized',
                                'weight_decay', 'split_num'])
np.random.seed(42) #locks random seed
skf = StratifiedKFold(n_splits=5)
for wd in weight_decay:
    archstr = "resnet18"
    for epochs in epoch_cycles:
        for lr in max_lr:
            test_name = archstr + '_' + str(epochs) + '_' + str(wd) + '_' + str(lr) + '_' + str('modified') + '_' + str('True')
            print(test_name)
            split_num = 1
            for train_index, val_index in skf.split(df_Test_Train.index, df_Test_Train['sex']):
                print(str(split_num))
                data_fold = (ImageList.from_df(df_Test_Train,'/home/jplineb/Chicken_Proj')
                            .split_by_idxs(train_index, val_index)
                            .label_from_df(cols='sex')
                            .transform(transforms, size=224)
                            .databunch(bs=2)).normalize()

                learn = cnn_learner(data_fold, models.resnet18, metrics=error_rate,
                                    callback_fns = [CSVLogger,AUROC, partial(EarlyStoppingCallback, monitor='AUROC', mode='max', min_delta=0.01, patience=100)], wd=wd)
                learn.fit_one_cycle(epochs, max_lr=lr)

                df_history = pd.read_csv('history.csv') # adds to dataframec created earlier (again norm and tsfms locked to ON)
                DFBig = DFBig.append({'test_name': test_name,
                                      'model_arch': 'resnet18',
                                      'transforms': 'modified',
                                      'normalized': 'true',
                                      'weight_decay': wd,
                                      'lr' :lr,
                                      'epoch': epochs,
                                      'split_num': split_num, # indicates kfold split
                                      'error_rate' : df_history.error_rate[epochs-1],
                                      'AUROC': df_history.AUROC[epochs-1],
                                      'train_df': df_history}, ignore_index = True)

                split_num+=1
                DFBig.to_csv('DF_new_hyperparameters.csv')
```

```
resnet18_5_0.01_0.01_modified_True
1
```

Total time: 00:13

epoch	train_loss	valid_loss	error_rate	AUROC	time
0	1.326058	0.746232	0.350000	0.640000	00:03
1	1.494292	0.928135	0.500000	0.700000	00:02
2	1.331969	1.452718	0.650000	0.450000	00:02
3	1.238274	1.205856	0.500000	0.430000	00:02
4	1.166000	0.877918	0.400000	0.510000	00:02

```
2
```

Total time: 00:11

Preview of the results.

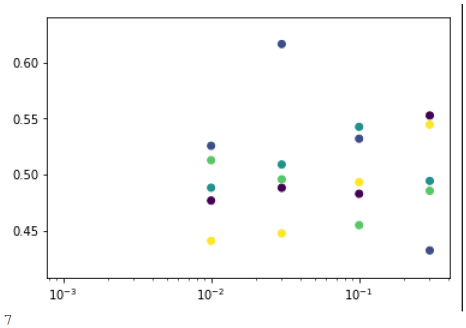
df_mean - DataFrame

Index	Unnamed: 0	normalized	weight_decay	split_num	AUROC	epoch	error_rate	lr
resnet18_7_1_0.1_modified_True	272	True	1	3	0.652346	7	0.447778	0.1
resnet18_7_1_0.3_modified_True	277	True	1	3	0.622247	7	0.447778	0.3
resnet18_9_0.3_0.3_modified_True	237	True	0.3	3	0.621654	9	0.425556	0.3
resnet18_5_0.03_0.03_modified_True	67	True	0.03	3	0.61637	5	0.433333	0.03
resnet18_7_0.03_0.1_modified_True	92	True	0.03	3	0.578556	7	0.445555	0.1
resnet18_9_1_0.03_modified_True	287	True	1	3	0.574148	9	0.476667	0.03
resnet18_7_0.1_0.01_modified_True	142	True	0.1	3	0.562519	7	0.501111	0.01
resnet18_7_0.1_0.3_modified_True	157	True	0.1	3	0.559901	7	0.466667	0.3
resnet18_9_0.3_0.01_modified_True	222	True	0.3	3	0.554321	9	0.451111	0.01
resnet18_5_0.01_0.3_modified_True	17	True	0.01	3	0.552753	5	0.447778	0.3
resnet18_7_0.01_0.03_modified_True	27	True	0.01	3	0.547383	7	0.464444	0.03
resnet18_5_1_0.3_modified_True	257	True	1	3	0.544617	5	0.501111	0.3
resnet18_5_0.1_0.1_modified_True	132	True	0.1	3	0.542642	5	0.478889	0.1
resnet18_5_0.03_0.1_modified_True	72	True	0.03	3	0.532099	5	0.446667	0.1
resnet18_7_1_0.03_modified_True	267	True	1	3	0.528642	7	0.49	0.03
resnet18_5_0.03_0.01_modified_True	62	True	0.03	3	0.525778	5	0.474444	0.01
resnet18_7_0.01_0.01_modified_True	22	True	0.01	3	0.519358	7	0.544444	0.01
resnet18_7_0.3_0.1_modified_True	212	True	0.3	3	0.51842	7	0.512222	0.1
resnet18_9_0.1_0.3_modified_True	177	True	0.1	3	0.516691	9	0.477778	0.3
resnet18_9_1_0.1_modified_True	292	True	1	3	0.513358	9	0.498889	0.1

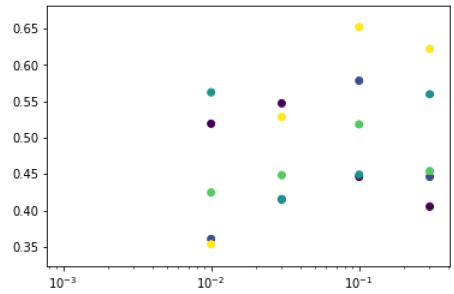
Format | Resize | Background color | Column min/max | Save and Close | Close

LR vs AUROC at each epoch level

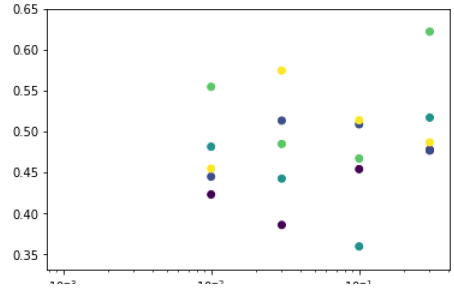
5



7



9



New hyperparameters Repeated Stratified K Fold

Test 1

- Resnet18
- 7 Epochs
- learning rate of 0.1
- weight decay of 1
- 5 Fold with 50 repeats

In [5]:
```
'''
This test resulted in 46.4% of the 250 AUROC scores being >=.5, this was not good enough
The standard deviation of AUROCS was .1308 with a mean of .4885
'''


max_lr = [0.1]
weight_decay = [1]
epoch_cycles = [7]
k=5
repeats = 50

transforms=[[RandTransform(tfm=TfmPixel (flip_lr), kwargs={}, p=0.5, resolved={}, do_run=True, is_random=True),
           RandTransform(tfm=TfmAffine (rotate), kwargs={'degrees': (-15.0, 15.0)}, p=0.75, resolved={}, do_run=True, is_random=True),
           RandTransform(tfm=TfmAffine (zoom), kwargs={'scale': (0.85, 1.15), 'row_pct': (0, 1), 'col_pct': (0, 1)}, p=0.5, resolved={}, do_run=True, is_random=True)],
            []]

# Creating Frame Work
DFBig = pd.DataFrame(columns = ['test_name','model_arch', 'transforms', 'normalized',
                                'weight_decay', 'split_num'])
np.random.seed(42) #locks random seed
rskf = RepeatedStratifiedKFold(n_splits=5, n_repeats = 50)
for wd in weight_decay:
    archstr = "resnet18"
    for epochs in epoch_cycles:
        for lr in max_lr:
            test_name = archstr + '_' + str(epochs) + '_' + str(wd) + '_' + str(lr) + '_' + str('modified') + '_' + str('True')
            print(test_name)
            split_num = 1
            for train_index, val_index in rskf.split(df_Test_Train.index, df_Test_Train['sex']):
                print(str(split_num))
                data_fold = (ImageList.from_df(df_Test_Train,'/home/jplineb/Chicken_Proj')
                            .split_by_idxs(train_index, val_index)
                            .label_from_df(cols='sex')
                            .transform(transforms, size=224)
                            .databunch(bs=2)).normalize()

                learn = cnn_learner(data_fold, models.resnet18, metrics=error_rate,
                                    callback_fns = [CSVLogger,AUROC, partial(EarlyStoppingCallback, monitor='AUROC', mode='max', min_delta=0.01, patience=100)], wd=wd)
                learn.fit_one_cycle(epochs, max_lr=lr)

                df_history = pd.read_csv('history.csv') # adds to dataframec created earlier (again norm and tsfms locked to ON)
                DFBig = DFBig.append({'test_name': test_name,
                                      'model_arch': 'resnet18',
                                      'transforms': 'modified',
                                      'normalized': 'true',
                                      'weight_decay': wd,
                                      'lr' :lr,
                                      'epoch': epochs,
                                      'split_num': split_num, # indicates kfold split
                                      'error_rate' : df_history.error_rate[epochs-1],
                                      'AUROC': df_history.AUROC[epochs-1],
                                      'train_df': df_history}, ignore_index = True)

                split_num+=1
                DFBig.to_csv('DF_resnet18_7_1_0.1_modified_True.csv')
```

resnet18_7_1_0.1_modified_True
1

Total time: 00:19

epoch	train_loss	valid_loss	error_rate	AUROC	time
0	1.321423	2.133742	0.500000	0.590000	00:02
1	1.211715	25.552029	0.500000	0.500000	00:02
2	1.432338	1.371010	0.500000	0.360000	00:02
3	1.351257	17248.728516	0.500000	0.500000	00:02
4	1.160453	0.857534	0.500000	0.620000	00:02
5	0.894236	0.699541	0.550000	0.610000	00:02
6	0.808082	0.692825	0.400000	0.550000	00:02

2

Preview of Results

df_2 - DataFrame

Index	Unnamed: 0	test_name	model_arch	transforms	normalized	weight_decay	split_num	AUROC	epoch	error_rate	lr	train_df
212	212	resnet1…	resnet18	modified	True	1	213	0.888889	7	0.222222	0.1	epoc…
106	106	resnet1…	resnet18	modified	True	1	107	0.81	7	0.5	0.1	epoc…
131	131	resnet1…	resnet18	modified	True	1	132	0.79	7	0.5	0.1	epoc…
181	181	resnet1…	resnet18	modified	True	1	182	0.79	7	0.45	0.1	epoc…
60	60	resnet1…	resnet18	modified	True	1	61	0.78	7	0.2	0.1	epoc…
194	194	resnet1…	resnet18	modified	True	1	195	0.765432	7	0.444444	0.1	epoc…
180	180	resnet1…	resnet18	modified	True	1	181	0.76	7	0.4	0.1	epoc…
122	122	resnet1…	resnet18	modified	True	1	123	0.740741	7	0.5	0.1	epoc…
74	74	resnet1…	resnet18	modified	True	1	75	0.728395	7	0.5	0.1	epoc…
116	116	resnet1…	resnet18	modified	True	1	117	0.72	7	0.35	0.1	epoc…
207	207	resnet1…	resnet18	modified	True	1	208	0.716049	7	0.5	0.1	epoc…
22	22	resnet1…	resnet18	modified	True	1	23	0.703704	7	0.5	0.1	epoc…
168	168	resnet1…	resnet18	modified	True	1	169	0.703704	7	0.277778	0.1	epoc…
55	55	resnet1…	resnet18	modified	True	1	56	0.7	7	0.4	0.1	epoc…
76	76	resnet1…	resnet18	modified	True	1	77	0.7	7	0.45	0.1	epoc…
79	79	resnet1…	resnet18	modified	True	1	80	0.691358	7	0.277778	0.1	epoc…
149	149	resnet1…	resnet18	modified	True	1	150	0.691358	7	0.333333	0.1	epoc…
159	159	resnet1…	resnet18	modified	True	1	160	0.691358	7	0.333333	0.1	epoc…
6	6	resnet1…	resnet18	modified	True	1	7	0.68	7	0.5	0.1	epoc…
5	5	resnet1…	resnet18	modified	True	1	6	0.67	7	0.4	0.1	epoc…
215	215	resnet1…	resnet18	modified	True	1	216	0.67	7	0.35	0.1	epoc…
59	59	resnet1…	resnet18	modified	True	1	60	0.666667	7	0.5	0.1	epoc…
84	84	resnet1…	resnet18	modified	True	1	85	0.666667	7	0.444444	0.1	epoc…
139	139	resnet1…	resnet18	modified	True	1	140	0.666667	7	0.388889	0.1	epoc…
172	172	resnet1…	resnet18	modified	True	1	173	0.666667	7	0.444444	0.1	epoc…
135	135	resnet1…	resnet18	modified	True	1	136	0.66	7	0.5	0.1	epoc…
235	235	resnet1…	resnet18	modified	True	1	236	0.66	7	0.5	0.1	epoc…

Format | Resize | Background color | Column min/max | Save and Close | Close

Test 2

- Resnet18
- 7 Epochs
- weight decay of 1
- learning rate of 0.3
- 5 Fold with 50 repeats

In [4]:
```
'''
This test resulted in 45.6% of the 250 AUROC scores being >=.5, this was not good enough
The standard deviation of AUROCS was .1354 with a mean of .4776434

'''

max_lr = [0.3]
weight_decay = [1]
epoch_cycles = [7]
k=5
secs_per_epoch=2

transforms=[[RandTransform(tfm=TfmPixel (flip_lr), kwargs={}, p=0.5, resolved={}, do_run=True, is_random=True),
           RandTransform(tfm=TfmAffine (rotate), kwargs={'degrees': (-15.0, 15.0)}, p=0.75, resolved={}, do_run=True, is_random=True),
           RandTransform(tfm=TfmAffine (zoom), kwargs={'scale': (0.85, 1.15), 'row_pct': (0, 1), 'col_pct': (0, 1)}, p=0.5, resolved={}, do_run=True, is_random=True)],
            []]

# Creating Frame Work
DFBig = pd.DataFrame(columns = ['test_name','model_arch', 'transforms', 'normalized',
                                'weight_decay', 'split_num'])
np.random.seed(42) #locks random seed
rskf = RepeatedStratifiedKFold(n_splits=5, n_repeats = 50)
for wd in weight_decay:
    archstr = "resnet18"
    for epochs in epoch_cycles:
        for lr in max_lr:
            test_name = archstr + '_' + str(epochs) + '_' + str(wd) + '_' + str(lr) + '_' + str('modified') + '_' + str('True')
            print(test_name)
            split_num = 1
            for train_index, val_index in rskf.split(df_Test_Train.index, df_Test_Train['sex']):
                print(str(split_num))
                data_fold = (ImageList.from_df(df_Test_Train,'/home/jplineb/Chicken_Proj')
                            .split_by_idxs(train_index, val_index)
                            .label_from_df(cols='sex')
                            .transform(transforms, size=224)
                            .databunch(bs=2)).normalize()

                learn = cnn_learner(data_fold, models.resnet18, metrics=error_rate,
                                    callback_fns = [CSVLogger,AUROC, partial(EarlyStoppingCallback, monitor='AUROC', mode='max', min_delta=0.01, patience=100)], wd=wd)
                learn.fit_one_cycle(epochs, max_lr=lr)

                df_history = pd.read_csv('history.csv') # adds to dataframec created earlier (again norm and tsfms locked to ON)
                DFBig = DFBig.append({'test_name': test_name,
                                      'model_arch': 'resnet18',
                                      'transforms': 'modified',
                                      'normalized': 'true',
                                      'weight_decay': wd,
                                      'lr' :lr,
                                      'epoch': epochs,
                                      'split_num': split_num, # indicates kfold split
                                      'error_rate' : df_history.error_rate[epochs-1],
                                      'AUROC': df_history.AUROC[epochs-1],
                                      'train_df': df_history}, ignore_index = True)

                split_num+=1
                DFBig.to_csv('DF_resnet18_7_1_0.3_modified_True.csv')
```

```
resnet18_7_1_0.3_modified_True
1
```

Total time: 00:15

epoch	train_loss	valid_loss	error_rate	AUROC	time
0	1.771569	1.175850	0.500000	0.510000	00:02
1	2.201258	913.106079	0.500000	0.500000	00:02
2	3.528614	10.479480	0.500000	0.500000	00:02
3	3.133857	7.218740	0.500000	0.805000	00:02
4	2.153565	0.723058	0.500000	0.420000	00:02
5	1.425225	0.697133	0.500000	0.440000	00:02
6	1.049895	0.696136	0.500000	0.520000	00:02

```
2
```

Preview of test results

df_2 - DataFrame

Index	Unnamed: 0	t_nai	del_a	nsfor	mali	weight_decay	split_num	AUROC	epoch	error_rate	lr	train_df
43	43	r...	r...	m...	T...	1	44	0.888889	7	0.5	0.3	epoc...
234	234	r...	r...	m...	T...	1	235	0.790123	7	0.5	0.3	epoc...
128	128	r...	r...	m...	T...	1	129	0.765432	7	0.5	0.3	epoc...
196	196	r...	r...	m...	T...	1	197	0.76	7	0.5	0.3	epoc...
199	199	r...	r...	m...	T...	1	200	0.753086	7	0.277778	0.3	epoc...
5	5	r...	r...	m...	T...	1	6	0.75	7	0.5	0.3	epoc...
57	57	r...	r...	m...	T...	1	58	0.740741	7	0.5	0.3	epoc...
69	69	r...	r...	m...	T...	1	70	0.740741	7	0.444444	0.3	epoc...
9	9	r...	r...	m...	T...	1	10	0.728395	7	0.333333	0.3	epoc...
139	139	r...	r...	m...	T...	1	140	0.728395	7	0.444444	0.3	epoc...
131	131	r...	r...	m...	T...	1	132	0.72	7	0.5	0.3	epoc...
94	94	r...	r...	m...	T...	1	95	0.716049	7	0.555556	0.3	epoc...
13	13	r...	r...	m...	T...	1	14	0.703704	7	0.333333	0.3	epoc...
195	195	r...	r...	m...	T...	1	196	0.7	7	0.5	0.3	epoc...
141	141	r...	r...	m...	T...	1	142	0.68	7	0.35	0.3	epoc...
36	36	r...	r...	m...	T...	1	37	0.67	7	0.5	0.3	epoc...
2	2	r...	r...	m...	T...	1	3	0.666667	7	0.5	0.3	epoc...
33	33	r...	r...	m...	T...	1	34	0.666667	7	0.5	0.3	epoc...
89	89	r...	r...	m...	T...	1	90	0.666667	7	0.555556	0.3	epoc...
109	109	r...	r...	m...	T...	1	110	0.666667	7	0.444444	0.3	epoc...
189	189	r...	r...	m...	T...	1	190	0.666667	7	0.5	0.3	epoc...
22	22	r...	r...	m...	T...	1	23	0.654321	7	0.5	0.3	epoc...
44	44	r...	r...	m...	T...	1	45	0.654321	7	0.5	0.3	epoc...
123	123	r...	r...	m...	T...	1	124	0.654321	7	0.444444	0.3	epoc...
19	19	r...	r...	m...	T...	1	20	0.641975	7	0.388889	0.3	epoc...
137	137	r...	r...	m...	T...	1	138	0.641975	7	0.5	0.3	epoc...
15	15	r...	r...	m...	T...	1	16	0.64	7	0.35	0.3	epoc...
95	95	r...	r...	m...	T...	1	96	0.63	7	0.4	0.3	epoc...
101	101	r...	r...	m...	T...	1	102	0.63	7	0.55	0.3	epoc...
106	106	r...	r...	m...	T...	1	107	0.63	7	0.4	0.3	epoc...
145	145	r...	r...	m...	T...	1	146	0.63	7	0.45	0.3	epoc...

Format Resize Background color Column min/max Save and Close Close