

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

import warnings
warnings.filterwarnings("ignore", category=UserWarning, module="torch.nn.functional")

from fastai.vision import ImageDataBunch, cnn_learner, ClassificationInterpretation
from fastai.metrics import error_rate
from pathlib import PosixPath
import numpy as np
```

URLs.PETS

↳ <https://s3.amazonaws.com/fast-ai-imageclas/oxford-iiit-pet>

```
data = untar_data(URLs.PETS)
```

↳ Downloading <https://s3.amazonaws.com/fast-ai-imageclas/oxford-iiit-pet.tgz>

```
data.ls()
```

↳ [PosixPath('/root/.fastai/data/oxford-iiit-pet/annotations'),  
PosixPath('/root/.fastai/data/oxford-iiit-pet/images')]

```
image_path = data/'images'
image_annotations = data/'annotations'
```

```
fnames = get_image_files(image_path)
```

```
fnames[:5]
```

↳ [PosixPath('/root/.fastai/data/oxford-iiit-pet/images/Bengal\_100.jpg'),  
PosixPath('/root/.fastai/data/oxford-iiit-pet/images/Persian\_34.jpg'),  
PosixPath('/root/.fastai/data/oxford-iiit-pet/images/miniature\_pinscher\_24.jpg'),  
PosixPath('/root/.fastai/data/oxford-iiit-pet/images/Persian\_98.jpg'),  
PosixPath('/root/.fastai/data/oxford-iiit-pet/images/staffordshire\_bull\_terri\_11.jpg')]

```
pat = r'([^\s]+)_\d+.jpg$'
```

Saving...

```
data_bunch = ImageDataBunch.from_name_re(fnames=fnames, path=image_path, pat=pat, bs=128)
```

```
data_bunch.device
```

↳ device(type='cuda')

```
data_bunch.show_batch(rows=3, figsize=(7,7))
```



american\_bulldog



american\_pit\_bull\_terrier



staffordshire\_bull\_terrier



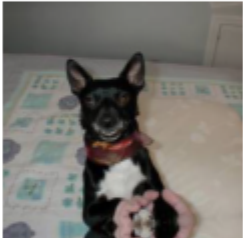
staffordshire\_bull\_terrier



havanese



chihuahua



pug



english\_setter



japanese\_chin



data\_bunch.classes



Saving...



```
[ 'Abyssinian',
  'Bengal',
  'Birman',
  'Bombay',
  'British_Shorthair',
  'Egyptian_Mau',
  'Maine_Coon',
  'Persian',
  'Ragdoll',
  ...]
```

```
len(data_bunch.classes), data_bunch.c
```

```
↳ (37, 37)
```

```
...american_pil_griffon ,
```

```
learn = cnn_learner(data=data_bunch, base_arch=models.resnet50, metrics=error_rate)
```

```
↳ Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /r
100% 97.8M/97.8M [00:00<00:00, 285MB/s]
```

```
...great_pyrenees ,
```

```
learn.model
```

```
↳
```

Saving...



```

Sequential(
  (0): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_s
    (2): ReLU(inplace=True)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=F
    (4): Sequential(
      (0): Bottleneck(
        (conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_rur
        (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1,
        (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_rur
        (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
        (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_ru
        (relu): ReLU(inplace=True)
        (downsample): Sequential(
          (0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
          (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_ru
        )
      )
    (1): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_rur
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_rur
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_ru
      (relu): ReLU(inplace=True)
    )
    (2): Bottleneck(
      (conv1): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1), bias=False
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_rur
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1,
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_rur
      (conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False
      (bn3): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_ru
      (relu): ReLU(inplace=True)
    )
  )
  (5): Sequential(
    (0): Bottleneck(
      (conv1): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1), bias=Fals
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_ru
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_ru
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=Fals
      (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_ru
      (relu): ReLU(inplace=True)
      (downsample): Sequential(
        (0): Conv2d(128, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_ru
      )
    )
    (1): Bottleneck(
      (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=Fals
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_ru
      (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(
      (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_ru
      (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=Fals
      (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_ru
      (relu): ReLU(inplace=True)
    )
  )
)

```

Saving...



```

)
(2): Bottleneck(
  (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
)
(3): Bottleneck(
  (conv1): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv3): Conv2d(128, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (bn3): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
)
(6): Sequential(
  (0): Bottleneck(
    (conv1): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (downsample): Sequential(
      (0): Conv2d(512, 1024, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (1): Bottleneck(
    (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
  )
  (2): Bottleneck(
    (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
  )
  (3): Bottleneck(
    (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
  )
  (4): Bottleneck(

```

Saving...



```

        (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
        (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
        (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
        (relu): ReLU(inplace=True)
    )
(5): Bottleneck(
  (conv1): Conv2d(1024, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
  (bn2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (conv3): Conv2d(256, 1024, kernel_size=(1, 1), stride=(1, 1), bias=False)
  (bn3): BatchNorm2d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace=True)
)
(7): Sequential(
  (0): Bottleneck(
    (conv1): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (conv3): Conv2d(512, 2048, kernel_size=(1, 1), stride=(1, 1), bias=False)
    (bn3): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace=True)
    (downsample): Sequential(
      (0): Conv2d(1024, 2048, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(2048, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
)

```

learn.metrics

↳ [<function fastai.metrics.error\_rate>]

```
(bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
```

learn.lr\_find()

↳  0.00% [0/1 00:00<00:00]

epoch	train_loss	valid_loss	error_rate	time
-------	------------	------------	------------	------

88	74.58%	[88/118 01:03<00:21 9.0187]
----	--------	-----------------------------

LR Finder is complete, type {learner\_name}.recorder.plot() to see the graph.

```

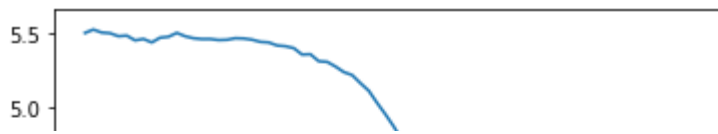
learn.recorder.plot()

```

learn.recorder.plot()

Saving...





```
learn.fit_one_cycle(8)
```

↗

epoch	train_loss	valid_loss	error_rate	time
0	0.937358	0.307451	0.094723	01:38
1	0.544735	0.323593	0.106901	01:36
2	0.439602	0.308359	0.097429	01:36
3	0.348346	0.329397	0.096076	01:35
4	0.260695	0.214930	0.061570	01:35
5	0.208083	0.212889	0.068336	01:35
6	0.160377	0.198017	0.059540	01:35
7	0.117305	0.192059	0.057510	01:35

```
learn.save('stage-1')
```

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

↗

```
loss, idx = interp.top_losses()
```

```
len(data_bunch.valid_ds)==len(loss)==len(idx)
```

↗ True

```
interp.plot_top_losses(16, figsize=(20,11))
```

↗

Saving...



Prediction/Actual/Loss/Probability

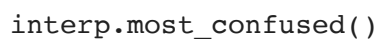


```
interp.plot_confusion_matrix(title='Pets Confusion Matrix', figsize=(20,20), dpi=60
```



Saving... X





Saving...



```
[('Ragdoll', 'Birman', 7),
 ('american_pit_bull_terrier', 'staffordshire_bull_terrier', 6),
 ('Egyptian_Mau', 'Bengal', 5),
 ('american_bulldog', 'boxer', 4),
 ('miniature_pinscher', 'chihuahua', 4),
 ('Bengal', 'Egyptian_Mau', 3),
 ('Birman', 'Ragdoll', 3),
 ('Birman', 'Siamese', 3),
 ('beagle', 'basset_hound', 3),
 ('Abyssinian', 'Bengal', 2),
 ('Ragdoll', 'Siamese', 2),
 ('Russian_Blue', 'Bombay', 2),
 ('Siamese', 'Birman', 2),
 ('american_pit_bull_terrier', 'american_bulldog', 2),
 ('english_cocker_spaniel', 'newfoundland', 2),
 ('english_setter', 'beagle', 2),
 ('shiba_inu', 'Abyssinian', 2),
 ('Bengal', 'pug', 1),
 ('British_Shorthair', 'Russian_Blue', 1),
 ('Egyptian_Mau', 'Russian_Blue', 1),
 ('Maine_Coon', 'Abyssinian', 1),
 ('Maine_Coon', 'Persian', 1),
 ('Maine_Coon', 'Ragdoll', 1),
 ('Persian', 'Bombay', 1),
 ('Ragdoll', 'British_Shorthair', 1),
 ('Ragdoll', 'Persian', 1),
 ('Russian_Blue', 'British_Shorthair', 1),
 ('Siamese', 'Ragdoll', 1),
 ('american_bulldog', 'staffordshire_bull_terrier', 1),
 ('american_pit_bull_terrier', 'boxer', 1),
 ('american_pit_bull_terrier', 'chihuahua', 1),
 ('american_pit_bull_terrier', 'miniature_pinscher', 1),
 ('beagle', 'american_bulldog', 1),
 ('boxer', 'american_bulldog', 1),
 ('boxer', 'beagle', 1),
 ('english_cocker_spaniel', 'german_shorthaired', 1),
 ('english_setter', 'english_cocker_spaniel', 1),
 ('keeshond', 'leonberger', 1),
 ('keeshond', 'pomeranian', 1),
 ('leonberger', 'newfoundland', 1),
 ('miniature_pinscher', 'american_pit_bull_terrier', 1),
 ('miniature_pinscher', 'staffordshire_bull_terrier', 1),
 ('pug', 'Bombay', 1),
 ('shiba_inu', 'samoyed', 1),
 ('staffordshire_bull_terrier', 'american_pit_bull_terrier', 1),
 ('wheaten_terrier', 'havanese', 1),
 ('wheaten_terrier', 'scottish_terrier', 1),
 ('yorkshire_terrier', 'Russian_Blue', 1)]
```

Saving...

