

Problem Set 2

Problem 2 - Fast.ai

Javier Palomares

Note: I executed this on Paperspace

Question 6

Download a NOT pre-trained resnet34 (read the Fast.ai documentation to see how to do this), and then by playing with the number of epochs and learning rates (possibly different learning rates across layers), see how low you can get the error. Can you get below 20%?

```
In [1]: %reload_ext autoreload
%autoreload 2
%matplotlib inline

In [2]: import torchvision.models as models
from fastai.vision import *
from fastai.metrics import error_rate
# batchSize
bs = 64
```

Load the pets data

```
In [4]: # download the pets data
path = untar_data(URLs.PETS); path

Out[4]: PosixPath('/storage/oxford-iiit-pet')

In [5]: # get the pets data
path_img = path/'images'
path_img

frames = get_image_files(path_img)
np.random.seed(2)
pet = re.compile(r'/([1-9])_id_\.jpg$')
data = ImageDataBunch.from_name_re(path_img, frames, pat, ds_tfms=get_transforms(), size=224, bs=bs)
data.normalize(ImageNetStats)

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



```
In [7]: print(data.classes)
len(data.classes),data.c

['Akita', 'Beagle', 'Basset Hound', 'Boxer', 'Chihuahua', 'English Cocker Spaniel', 'English Setter', 'German Shorthaired', 'Great Pyrenees', 'Havanese', 'Japanese Chin', 'Keeshond', 'Leonberger', 'Miniature Pinscher', 'Newfoundland', 'Pomeranian', 'Pug', 'Saint Bernard', 'Samoyed', 'Scottish Terrier', 'Shiba Inu', 'Staffordshire Bull Terrier', 'Weimaraner', 'Yorkshire Terrier']

Out[7]: (37, 37)
```

Training: Resnet34 that has not been pretrained

```
In [14]: %reload_ext autoreload
%autoreload 2
%matplotlib inline

In [14]: #download a resnet34 model that has not been pre trained
learn = create_cnn(data.models.resnet34, metrics=error_rate, pretrained=False)
```

In [15]: learn.model

```
Out[15]: Sequential
(0): Sequential
  (0): Conv2d(1, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
  (w1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (relu): ReLU(inplace)
  (1): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  (4): Sequential
    (0): BasicBlock
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    }
    (1): BasicBlock
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    }
    (2): BasicBlock
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    }
  }
(5): Sequential
  (0): BasicBlock
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
    (downsample): Sequential
      (0): Conv2d(128, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    }
  }
  (1): BasicBlock
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
  (2): BasicBlock
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
  (3): BasicBlock
    (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
}
(6): Sequential
  (0): BasicBlock
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
    (downsample): Sequential
      (0): Conv2d(256, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    }
  }
  (1): BasicBlock
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
  (2): BasicBlock
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
  (3): BasicBlock
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
  (4): BasicBlock
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
  (5): BasicBlock
    (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (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)
  }
}
(7): Sequential
  (0): BasicBlock
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (downsample): Sequential
      (0): Conv2d(512, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
      (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    }
  }
  (1): BasicBlock
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  }
  (2): BasicBlock
    (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (relu): ReLU(inplace)
    (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
    (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  }
}
(8): Sequential
  (0): AdaptiveConcatPool2d()
  (w): AdaptiveAvgPool2d(output_size=1)
  (w): AdaptiveMaxPool2d(output_size=1)
}
(1): Flatten()
(2): BatchNorm1d(1824, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): Dropout(p=0.2)
(4): Linear(in_features=1824, out_features=512, bias=True)
(5): ReLU(inplace)
(6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(7): Dropout(p=0.5)
(8): Linear(in_features=512, out_features=37, bias=True)
}
```

In [26]: learn.fit_one_cycle(4)

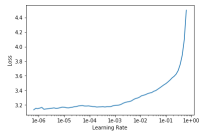
Total time: 01:08

epoch	train_loss	valid_loss	error_rate
1	3.269172	2.766287	0.339167
2	1.865421	1.393417	0.205129
3	1.433341	1.129422	0.260623
4	1.246876	1.019161	0.22027

In [27]: learn.lr_find()

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

In [28]: learn.recorder.plot()



In [28]: learn.unfreeze()

learn.fit_one_cycle(4,max_lr=llor(1e-6,1e-4))

Total time: 01:55

epoch	train_loss	valid_loss	error_rate
1	3.141590	2.997653	0.319350
2	1.110278	2.872866	0.319350
3	3.088112	2.848878	0.307172
4	3.086308	2.844144	0.313261

In [23]: learn.save('stage-1')

In [26]:

learn.load('stage-1')

Out[26]: learner(data=ImageDataLunch;

```

Train: LabelList
y: CategoryList (5912 items)
[Category english_cocker_spaniel, Category shiba_inu, Category Siamese, Category yorkshire_terrier, Category american_building]...
Path: /storage/oxford-11it-pet/images
x: ImageList (5912 items)
[Image (3, 332, 580), Image (3, 375, 580), Image (3, 580, 375), Image (3, 375, 580), Image (3, 334, 580)]...
Path: /storage/oxford-11it-pet/images;

Valid: LabelList
y: CategoryList (1478 items)
[Category Japanese_chin, Category Russian_Blue, Category scottish_terrier, Category Bengal, Category german_shorthaired]...
Path: /storage/oxford-11it-pet/images
x: ImageList (1478 items)
[Image (3, 580, 410), Image (3, 332, 580), Image (3, 332, 580), Image (3, 380, 380), Image (3, 334, 580)]...
Path: /storage/oxford-11it-pet/images;

```

```

Test: None, model=Sequential(
  (0): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(1, 1), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  )
  (4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (2): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (5): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (2): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (3): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
  )
  (6): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (2): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (3): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (4): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (5): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
  )
  (7): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (2): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (8): AdaptiveAvgPool2d(output_size=1)
  (9): AdaptiveMaxPool2d(output_size=1)
  (1): Flatten()
  (2): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (3): Dropout(p=0.25)
  (4): Linear(in_features=1024, out_features=512, bias=True)
  (5): ReLU(inplace)
  (6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (7): Dropout(p=0.5)
  (8): Linear(in_features=512, out_features=7, bias=True)
), opt_func=torch.nn.parallel.cnnutils.Adam.Adam, beta=(0.9, 0.99), loss_func=FlattenLoss of CrossEntropyLoss(), metrics=(function error_rate at 0x7f7a773180b8), true_val=True, bn_val=True, wd=0.01, train_bn=True, path_posix_path('/storage/oxford-11it-pet/images'), model_dir='models', callback_func=<class 'Fastai.basic_train_recorder'>), callback=

```

```
(15): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(16): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(17): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(18): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(19): ReLU(inplace)
(20): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(21): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(22): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(23): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(24): ReLU(inplace)
(25): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(26): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(27): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(28): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(29): ReLU(inplace)
(30): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(31): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(32): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(33): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(34): ReLU(inplace)
(35): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(36): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(37): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(38): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(39): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(40): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(41): ReLU(inplace)
(42): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(43): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(44): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(45): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(46): ReLU(inplace)
(47): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(48): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
), Sequential(
  (0): AdaptiveAvgPool2d(output_size=1)
  (1): AdaptiveMaxPool2d(output_size=1)
  (2): Flatten()
  (3): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (4): Dropout(p=0.25)
  (5): Linear(in_features=1024, out_features=512, bias=True)
  (6): ReLU(inplace)
  (7): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (8): Dropout(p=0.5)
  (9): Linear(in_features=512, out_features=37, bias=True)
))

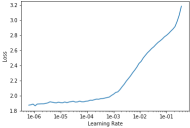
In [27]: learner.fit_one_cycle(8,max_lr=1e-04(lr=4,lr=2))

Total time: 03:50

epoch  train_loss  valid_loss  error_rate
1  3.170766  2.123318  0.876194
2  3.267846  3.346110  0.980419
3  3.119638  2.039588  0.864644
4  2.877096  1.266974  0.581827
5  2.642141  2.547389  0.730941
6  2.405128  2.235546  0.807718
7  2.187629  2.021127  0.819829
8  2.022288  1.928837  0.892016

In [28]: learner.lr_find()
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.

In [29]: learner.recorder.plot()



In [30]: learner.save('stage-2')
# why do my plots look like this?

In [31]: learner.fit_one_cycle(8,max_lr=1e-04(lr=4,lr=3))

Total time: 03:49

epoch  train_loss  valid_loss  error_rate
1  1.948058  1.974946  0.596752
2  2.079157  2.267982  0.664263
3  2.060719  2.115646  0.635318
4  1.940248  2.225805  0.821786
5  1.786113  1.783409  0.548391
6  1.625952  1.644986  0.508119
7  1.602997  1.497387  0.464688
8  1.438241  1.464517  0.462628

In [32]: learner.save('stage-3')
```

```
In [44]: learn.load('stage-3')
```

Out[44]: learner(data=ImageDataLunch;

```

Train: LabelList
y: CategoryList (5912 items)
[Category english_cocker_spaniel, Category shiba_inu, Category Siamese, Category yorkshire_terrier, Category american_building]...
Path: /storage/oxford-111t-pet/images
x: ImageList (5912 items)
[Image (3, 332, 500), Image (3, 332, 500), Image (3, 500, 375), Image (3, 375, 500), Image (3, 334, 500)]...
Path: /storage/oxford-111t-pet/images;

Valid: LabelList
y: CategoryList (1478 items)
[Category Japanese_chin, Category Russian_Blue, Category scottish_terrier, Category Bengal, Category german_shorthaired]...
Path: /storage/oxford-111t-pet/images
x: ImageList (1478 items)
[Image (3, 500, 410), Image (3, 333, 500), Image (3, 332, 500), Image (3, 300, 300), Image (3, 334, 500)]...
Path: /storage/oxford-111t-pet/images;

```

```

Test: None, model=Sequential(
  (0): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(1, 1), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  )
  (4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (2): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (5): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (2): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (3): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
  )
  (6): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (2): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (3): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (4): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (5): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
  )
  (7): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (2): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (8): AdaptiveAvgPool2d(output_size=1)
  (9): AdaptiveAvgPool2d(output_size=1)
  (1): Flatten()
  (2): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (3): Dropout(p=0.25)
  (4): Linear(in_features=1024, out_features=512, bias=True)
  (5): ReLU(inplace)
  (6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (7): Dropout(p=0.5)
  (8): Linear(in_features=512, out_features=7, bias=False)
), opt_func=torch.nn.parallel.cnnutils.Adam.Adam, beta=(0.9, 0.99), loss_func=FlattenLoss of CrossEntropyLoss(), metrics=(function error_rate at 0x7f7a773180b8), true_val=True, bn_val=True, wd=0.01, train_bn=True, path=PosixPath('/storage/oxford-111t-pet/images'), model_dir='models', callback_func=[class 'Fastai.basic_train_recorder'], callback=

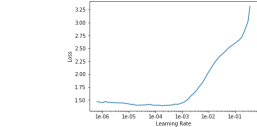
```

```
(15): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(16): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(17): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(18): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(19): ReLU(inplace)
(20): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(21): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(22): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(23): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(24): ReLU(inplace)
(25): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(26): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(27): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(28): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(29): ReLU(inplace)
(30): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(31): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(32): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(33): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(34): ReLU(inplace)
(35): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(36): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(37): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(38): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(39): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(40): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(41): ReLU(inplace)
(42): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(43): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(44): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(45): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(46): ReLU(inplace)
(47): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(48): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
), Sequential(
  (0): AdaptiveAvgPool2d(output_size=1)
  (1): AdaptiveMaxPool2d(output_size=1)
  (2): Flatten()
  (3): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (4): Dropout(p=0.25)
  (5): Linear(in_features=1024, out_features=512, bias=True)
  (6): ReLU(inplace)
  (7): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (8): Dropout(p=0.5)
  (9): Linear(in_features=512, out_features=37, bias=True)
))
```

In [45]: learner.lr_find()

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

In [46]: learner.recorder.plot()



In [47]: learner.unfreeze()
learner.fit_one_cycle(8,max_lr=11ice(1e-6,1e-4))
Total time: 02:50

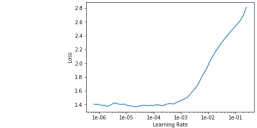
epoch	train_loss	valid_loss	error_rate
1	1.389746	1.465220	0.448579
2	1.385610	1.454856	0.451286
3	1.384186	1.448791	0.447903
4	1.384853	1.431807	0.441813
5	1.383073	1.426069	0.439107
6	1.389131	1.424824	0.439793
7	1.381168	1.424801	0.439107
8	1.385599	1.422893	0.437754

In [48]: learner.lr_find()

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

In [49]: learner.save('stage-4')

In [50]: learner.recorder.plot()




```
In [54]: learn.load('stage-4')
```

Out[54]: learner(data=ImageDataLunch;

```

Train: LabelList
y: CategoryList (5912 items)
[Category english_cocker_spaniel, Category shiba_inu, Category Siamese, Category yorkshire_terrier, Category american_building]...
Path: /storage/oxford-11it-pet/images
x: ImageList (5912 items)
[Image (3, 332, 580), Image (3, 332, 580), Image (3, 580, 375), Image (3, 375, 580), Image (3, 334, 580)]...
Path: /storage/oxford-11it-pet/images;

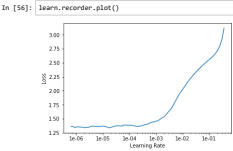
Valid: LabelList
y: CategoryList (1478 items)
[Category Japanese_chin, Category Russian_Blue, Category scottish_terrier, Category Bengal, Category german_shorthaired]...
Path: /storage/oxford-11it-pet/images
x: ImageList (1478 items)
[Image (3, 580, 410), Image (3, 332, 580), Image (3, 332, 580), Image (3, 380, 380), Image (3, 334, 580)]...
Path: /storage/oxford-11it-pet/images;

Test: None, model=Sequential(
  (0): Sequential(
    (0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(1, 1), bias=False)
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
  )
  (4): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (1): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (2): BasicBlock(
      (conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (5): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
      (downsample): Sequential(
        (0): Conv2d(64, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (2): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (3): BasicBlock(
      (conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
  )
  (6): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
      (downsample): Sequential(
        (0): Conv2d(128, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (2): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (3): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (4): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
    (5): BasicBlock(
      (conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (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)
    )
  )
  (7): Sequential(
    (0): BasicBlock(
      (conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (downsample): Sequential(
        (0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
        (1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      )
    )
    (1): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
    (2): BasicBlock(
      (conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
      (relu): ReLU(inplace)
      (conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
      (bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    )
  )
  (8): Sequential(
    (0): AdaptiveConvPool2d(output_size=1)
    (mp): AdaptiveAvgPool2d(output_size=1)
  )
  (9): Flatten()
  (10): BatchNorm1d(1824, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (11): Dropout(p=0.25)
  (12): Linear(in_features=1824, out_features=512, bias=True)
  (13): ReLU(inplace)
  (14): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (15): Dropout(p=0.5)
  (16): Linear(in_features=512, out_features=7, bias=True)
  )
  ), opt_func=torch.nn.parallel.cnnutils.Adam.Adam, beta=(0.9, 0.99), loss_func=FlattenLoss of CrossEntropyLoss(), metrics=(function error_rate at 0x7f7a773180b8), true_val=True, bn_val=True, wd=0.01, train_bn=True, path_posix_path('/storage/oxford-11it-pet/images'), model_dir='models', callback_func=<class 'Fastai.basic_train_recorder'>), callback=

```

```
(15): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(16): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(17): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(18): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(19): ReLU(inplace)
(20): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(21): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(22): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(23): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(24): ReLU(inplace)
(25): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(26): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(27): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(28): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(29): ReLU(inplace)
(30): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(31): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(32): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(33): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(34): ReLU(inplace)
(35): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(36): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(37): Conv2d(512, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(38): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(39): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(40): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(41): ReLU(inplace)
(42): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(43): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(44): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(45): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(46): ReLU(inplace)
(47): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(48): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
), Sequential(
  (0): AdaptiveAvgPool2d(output_size=1)
  (1): AdaptiveMaxPool2d(output_size=1)
  (2): Flatten
  (3): BatchNorm1d(1804, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (4): Dropout(p=0.25)
  (5): Linear(in_features=1804, out_features=512, bias=True)
  (6): ReLU(inplace)
  (7): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (8): Dropout(p=0.5)
  (9): Linear(in_features=512, out_features=37, bias=True)
))
```

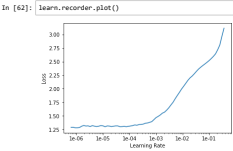
In [55]: learner.lr_find()
LR finder is complete, type learner.lr_recorder.plot() to see the graph.



In [57]: learner.fit_one_cycle(3,max_lr=slice(1e-6,1e-4))
Total time: 01:26

epoch	train_loss	valid_loss	error_rate
1	1.336868	1.419860	0.435947
2	1.337058	1.416310	0.436401
3	1.320803	1.416808	0.430311

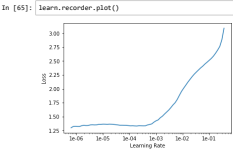
In [61]: learner.lr_find()
LR finder is complete, type learner.lr_recorder.plot() to see the graph.



In [63]: learner.fit_one_cycle(3,max_lr=slice(1e-6,5e-5))
Total time: 01:26

epoch	train_loss	valid_loss	error_rate
1	1.345208	1.420546	0.439783
2	1.320977	1.412795	0.439187
3	1.338790	1.412673	0.439753

In [64]: learner.lr_find()
LR finder is complete, type learner.lr_recorder.plot() to see the graph.



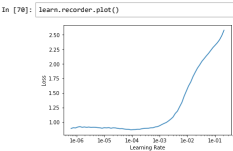
In [66]: learner.fit_one_cycle(10,max_lr=slice(1e-6,5e-3))
Total time: 04:50

epoch	train_loss	valid_loss	error_rate
1	1.389756	1.665010	0.492557
2	1.617047	2.318883	0.641487
3	1.776964	2.801384	0.680158
4	1.731678	2.202461	0.510889
5	1.628920	1.629454	0.460488
6	1.478357	1.563891	0.473928
7	1.289244	1.494651	0.447228
8	1.087861	1.177820	0.365339
9	0.981871	1.084234	0.341001
10	0.887030	1.075895	0.338295

In [67]: learner.save('stage-5')

The error rate is coming down, fit by running 10 epochs again

In [69]: learner.lr_find()
LR finder is complete, type learner.lr_recorder.plot() to see the graph.



In [71]: learner.fit_one_cycle(10,max_lr=slice(1e-6,5e-3))
Total time: 04:48

epoch	train_loss	valid_loss	error_rate
1	0.930648	1.162842	0.382652
2	1.095850	1.750707	0.602030
3	1.290322	1.648293	0.506089
4	1.293823	1.763878	0.495264
5	1.207166	1.449262	0.432341
6	1.081000	1.333827	0.399885
7	0.892772	1.172849	0.368981
8	0.784106	0.985890	0.305442
9	0.685142	0.948558	0.297780
10	0.622711	0.942571	0.284317

In [72]: learner.save('stage-5')

The error rate is going down, fit for some more

In [73]: learner.lr_find()
LR finder is complete, type learner.lr_recorder.plot() to see the graph.

In [74]:

```
learn.reorder_plot()
```



In [75]:

```
learn.fit_one_cycle(10,max_lr=slice(1e-4,5e-3))
```

Total time: 04:47

epoch	train_loss	valid_loss	error_rate
1	0.640262	1.109783	0.344384
2	0.607246	1.373836	0.397835
3	0.996260	1.373897	0.414073
4	1.012437	1.407689	0.408660
5	0.954484	1.391077	0.410680
6	0.867274	1.168899	0.348444
7	0.753185	1.083033	0.326793
8	0.612640	0.959522	0.286188
9	0.605453	0.923834	0.278049
10	0.462423	0.916791	0.278766

In [76]:

```
learn.lr_find()
```

LR Finder is complete, type (learner_name).reorder_plot() to see the graph.

In [77]:

```
learn.reorder_plot()
```



Fit run a few more epochs

In [78]:

```
learn.fit_one_cycle(5,max_lr=slice(1e-5,1e-4))
```

Total time: 02:23

epoch	train_loss	valid_loss	error_rate
1	0.448540	0.913802	0.277402
2	0.458828	0.913210	0.276049
3	0.445356	0.912334	0.271313
4	0.438490	0.900774	0.268283
5	0.423028	0.897274	0.270636

In [79]:

```
learn.save('stage-6')
```

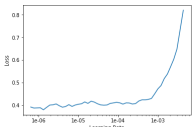
In [80]:

```
learn.lr_find()
```

LR Finder is complete, type (learner_name).reorder_plot() to see the graph.

In [81]:

```
learn.reorder_plot()
```



In [82]:

```
learn.fit_one_cycle(10,max_lr=slice(1e-5,5e-3))
```

Total time: 04:47

epoch	train_loss	valid_loss	error_rate
1	0.417322	0.906146	0.288808
2	0.447468	0.983210	0.289108
3	0.488001	1.098195	0.308825
4	0.583346	1.067062	0.287581
5	0.603938	1.002879	0.292863
6	0.453988	1.014915	0.295851
7	0.403970	0.966329	0.274019
8	0.355747	0.944091	0.267930
9	0.326643	0.971328	0.269959
10	0.317467	0.957206	0.264647

In [83]:

```
learn.save('stage-6')
```

The error isn't decreasing much now.

In [84]:

```
learn.lr_find()
```

LR Finder is complete, type (learner_name).reorder_plot() to see the graph.

In [85]:

```
learn.reorder_plot()
```



In [86]:

```
learn.fit_one_cycle(5,max_lr=slice(1e-5,1e-4))
```

Total time: 02:24

epoch	train_loss	valid_loss	error_rate
1	0.295457	0.956629	0.263194
2	0.302348	0.968988	0.273342
3	0.286331	0.971627	0.263879
4	0.280742	0.957635	0.262617
5	0.286637	0.958174	0.259134

In [87]:

```
learn.lr_find()
```

LR Finder is complete, type (learner_name).reorder_plot() to see the graph.

In [88]:

```
learn.reorder_plot()
```



In [89]:

```
learn.fit_one_cycle(10,max_lr=slice(1e-5,1e-4))
```

Total time: 04:40

epoch	train_loss	valid_loss	error_rate
1	0.277758	0.957766	0.259811
2	0.288726	0.957216	0.257781
3	0.285266	0.977196	0.265223
4	0.291110	0.968199	0.270636
5	0.272348	0.947386	0.263194
6	0.275718	0.969199	0.265223
7	0.291499	0.964336	0.263194
8	0.268800	0.966605	0.267253
9	0.276111	0.965372	0.271989
10	0.287614	0.953666	0.268980

https://nppuxajt.gradient.paperspace.com/nbconvert/html/Data_Mining/Data_Mining_Hw2_Problem_2.ipynb?download=false

12/15

I'm not seeing the error rate come down from around .26, and don't expect that running more epochs even at different rates would get the error rate under .2

Question 7

Instructions

And for the main part of this exercise: download (and label) your own data set of your choice, create a classification problem, and then use the main bodi/ideas of this notebook to build a classifier. It does not need to be a multi-label classifier.

Solution

I created a dataset by downloading images using Bing Search API. I found a good article that explained how to do this at <https://www.courmagnard.com/2018/04/06/how-to-quickly-build-a-deep-learning-image-dataset/>. I created a separate python script to create my data set: create_dataset.py and used it to download approximately 850 images each of Pikachu and Squirtle that I have zipped up and uploaded to Paperspace. I will train a classification model on this dataset.

```
In [1]: import zipfile
# expand the zip
dataset_file = "/storage/dataset.zip"
zip_ref = zipfile.ZipFile(dataset_file, 'r')
zip_ref.extractall("/storage/dataset")
zip_ref.close()
```

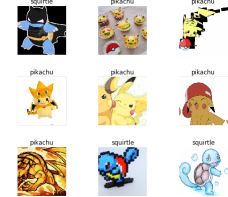
```
In [1]: %reload_ext autoreload
%autoreload 2
%matplotlib inline
import torchvision.models as models
from fastai.vision import *
from fastai.metrics import error_rate
# torchvision
bs = 64
```

Loading the Data

I'll load the data from the dataset I built, and set 20% of the data to be chosen for validation

```
In [2]: data = ImageDataBunch.from_folder(path="/storage/dataset", valid_pct=0.2, bs=bs, size=224, ds_tfms=get_transforms(), normalize=ImageNet_stats)
```

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



```
In [4]: print(data.classes)
len(data.classes), data.c
['pikachu', 'squirtle']
```

```
Out[4]: (2, 2)
```

So far the data load looks good. There are 2 classes for the images "Squirtle, and Pikachu"

```
In [4]: np.random.seed(2)
```

Training the model

I'll use a pretrained resnet34 to train on my dataset

```
In [4]: learn = create_cnn(data, models.resnet34, metrics=error_rate)
```

In [5]: learn.model

```

Out[5]: Sequential
(0): Sequential
(0): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
(01): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(1): ReLU(inplace)
(11): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
(4): Sequential
(0): BasicBlock
(conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(1): BasicBlock
(conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(2): BasicBlock
(conv1): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(5): Sequential
(0): BasicBlock
(conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
(downsample): Sequential
(0): Conv2d(128, 128, kernel_size=(1, 1), stride=(2, 2), bias=False)
(1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(1): BasicBlock
(conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(2): BasicBlock
(conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(3): BasicBlock
(conv1): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(6): Sequential
(0): BasicBlock
(conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
(downsample): Sequential
(0): Conv2d(256, 256, kernel_size=(1, 1), stride=(2, 2), bias=False)
(1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(1): BasicBlock
(conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(2): BasicBlock
(conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(3): BasicBlock
(conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(4): BasicBlock
(conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(5): BasicBlock
(conv1): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(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)
}
(7): Sequential
(0): BasicBlock
(conv1): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(downsample): Sequential
(0): Conv2d(256, 512, kernel_size=(1, 1), stride=(2, 2), bias=False)
(1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(1): BasicBlock
(conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(2): BasicBlock
(conv1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn1): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(relu): ReLU(inplace)
(conv2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
}
(1): Sequential
(0): AdaptiveConcatPool2d
(m): AdaptiveAvgPool2d(output_size=1)
(m): AdaptiveMaxPool2d(output_size=1)
}
(1): Flatten
(2): BatchNorm1d(1824, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(3): Dropout(p=0.2)
(4): Linear(in_features=1824, out_features=512, bias=True)
(5): ReLU(inplace)
(6): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
(7): Dropout(p=0.5)
(8): Linear(in_features=512, out_features=2, bias=True)
}

```

In [6]: learn.fit_one_cycle(4)

Total time: 01:51

epoch	train_loss	valid_loss	error_rate
1	0.46946	0.19805	0.26468
2	0.30478	0.16202	0.26468
3	0.30323	0.128261	0.26468
4	0.239137	0.12097	0.046875

Even after just 4 epochs, the error rate is already very low

In [7]: learn.save('stage-1')

In [8]: intep = ClassificationInterpretation.from_learner(learn)

losses_idx = intep.top_losses()

len(data.valid_dls) - len(losses) == len(losses)

Out[8]: True

In [9]: intep.plot_top_losses(9, figsize=(15,11))

prediction/actual/loss/probability

pikachuquette / 1.47 / 0.01



pikachuquette / 3.92 / 0.02



pikachuquette / 2.37 / 0.10



squirtlepkachu / 1.88 / 0.15



squirtlepkachu / 1.87 / 0.15



squirtlepkachu / 1.84 / 0.16



pikachuquette / 1.76 / 0.17



pikachuquette / 1.60 / 0.20



squirtlepkachu / 1.11 / 0.13

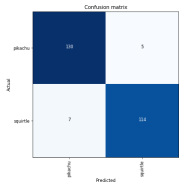


In [18]:

doc(interp_plot_top_losses)

In [13]:

interp_plot_confusion_matrix(figsize=(6,6), dpi=60)



Confusion matrix

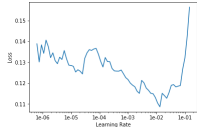
	Actual picture	Actual mouth
Predicted picture	111	5
Predicted mouth	7	104

In [14]:

learn.lr_find()
LR Finder is complete, type {learner_name}.reorder_plot() to see the graph.

In [15]:

learn.reorder_plot()



Loss

Learning Rate

In [16]:

learn.fit_one_cycle(2, max_lr=slice(1e-6, 5e-3))
Total time: 00:52

epoch	train_loss	valid_loss	error_rate
1	0.154032	0.150688	0.035168
2	0.121907	0.120194	0.030962

The model is very accurate!