```
from fastai import *
from fastai.vision import *
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

```
path = Path('drive/My Drive')
```

```
path.ls()
```

```
dest = path/'bears/teddy'
```

```
dest.ls()
```

```
file = 'bears/teddy/teddys.txt'
```

```
download_images(urls=path/file, dest=dest, max_pics=300)
```

```
classes = ['teddy','grizzly','black']
```

```
mod_path = path/'bears'
```

```
for c in classes:
    print(c)
    verify_images(mod_path/c, delete=True, max_size=500)
```

Saved successfully!

```
bears_path = path/'bears'
```

```
np.random.seed(42)
data = ImageDataBunch.from_folder(bears_path, train=".", valid_pct=0.2,
        ds_tfms=get_transforms(), size=224, num_workers=4).normalize(imagenet_stats)
```

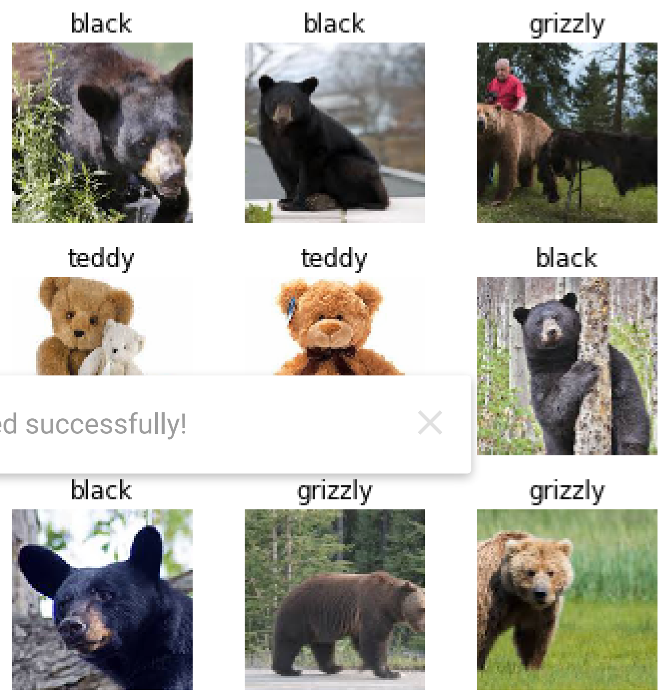
```
data
```

```
ImageDataBunch;

Train: LabelList (305 items)
x: ImageList
Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image
y: CategoryList
black,black,black,black,black
Path: drive/My Drive/bears;

Valid: LabelList (76 items)
x: ImageList
Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image (3, 224, 224),Image
y: CategoryList
grizzly,grizzly,black,black,black
Path: drive/My Drive/bears;

Test: None
```

```
data.classes
```

```
['black', 'grizzly', 'teddy', 'testing']
```

```
data.show_batch(rows=3, figsize=(5,5))
```



Saved successfully!

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

```
learn.fit_one_cycle(4)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.419616	0.135103	0.039474	00:06
1	0.791952	0.457623	0.052632	00:04
2	0.564850	0.437018	0.052632	00:04
3	0.461121	0.378157	0.052632	00:04

```
learn.save('bears_stage_1')
```

```
learn.unfreeze()
```

```
learn.lr_find()
```

56.00% [14/25 00:53<00:42]

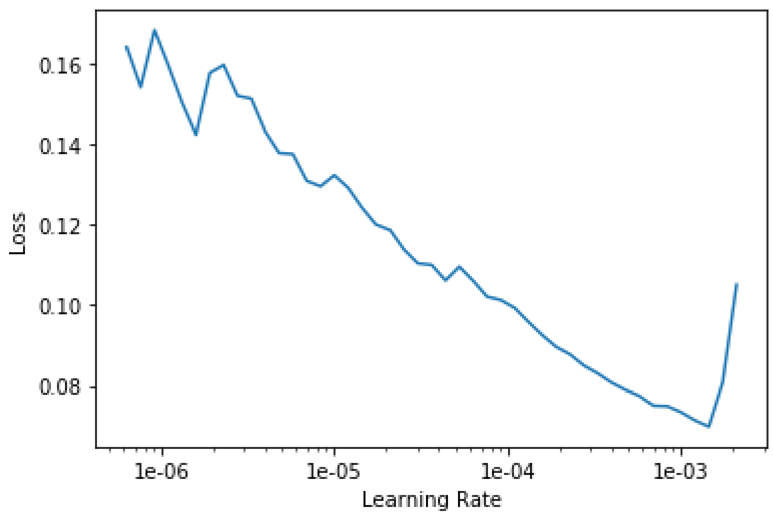
epoch	train_loss	valid_loss	error_rate	time
0	0.170179	#na#	00:04	
1	0.184039	#na#	00:03	
2	0.154112	#na#	00:04	
3	0.142315	#na#	00:03	
4	0.151273	#na#	00:03	
5	0.130930	#na#	00:03	
6	0.124237	#na#	00:03	
7	0.110373	#na#	00:03	
[illegible]	[illegible]	[illegible]	00:03	
[illegible]	[illegible]	[illegible]	00:03	
10	0.084986	#na#	00:03	
11	0.077458	#na#	00:03	
12	0.071370	#na#	00:03	
13	0.187966	#na#	00:03	

Saved successfully! ×

75.00% [3/4 00:03<00:01 0.2718]

[illegible]

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



```
learn.load('bears_stage_1')
learn.fit_one_cycle(2, max_lr=slice(3e-5, 3e-4))
```

epoch	train_loss	valid_loss	error_rate	time
0	0.122900	0.234929	0.052632	00:04
1	0.101325	0.228041	0.039474	00:04

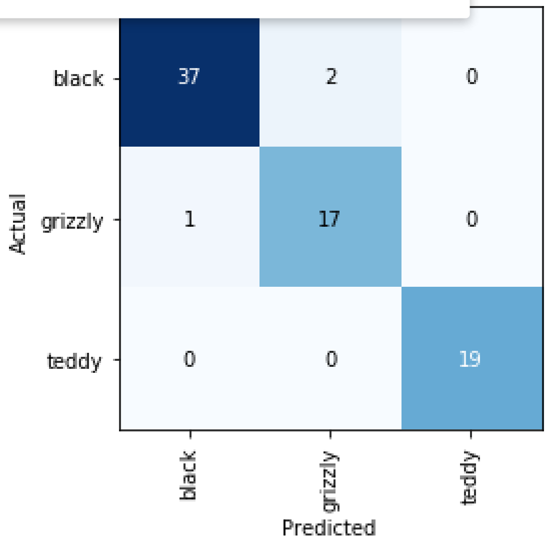
```
learn.save('bears_stage_2')
```

```
learn.load('bears_stage_2')
```

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

```
interpretation.plot_confusion_matrix()
```

Saved successfully!



```
path
```

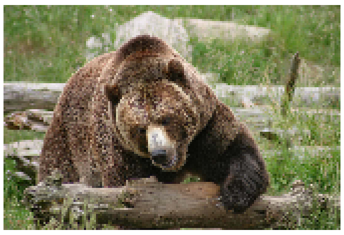
```
    PosixPath('drive/My Drive')
```

```
bears_path
```

```
    PosixPath('drive/My Drive/bears')
```

```
img = open_image(bears_path/'testing/Grizzly-bear_test.jpg')
```

```
img.show(size=(5,5))
```



```
classes = ['black', 'grizzly', 'teddy']
```

```
data2 = ImageDataBunch.single_from_classes(bears_path, classes, ds_tfms=get_transforms(),
```

```
learn = cnn_learner(data2, models.resnet50)
```

```
learn.load('bears_stage_2')
```

```
pred_class, pred_idx, pred_outputs = learn.predict(img)
```

```
pred_class
```

```
    Category grizzly
```

Saved successfully!

Very high learning rate

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

```
    Downloading: "https://download.pytorch.org/models/resnet50-19c8e357.pth" to /root/.ca
    100%|██████████| 97.8M/97.8M [00:00<00:00, 232MB/s]
```

```
learn.fit_one_cycle(1, max_lr=0.5)
```

epoch	train_loss	valid_loss	error_rate	time
0	18.309895	1235156992000.000000	1.000000	00:06

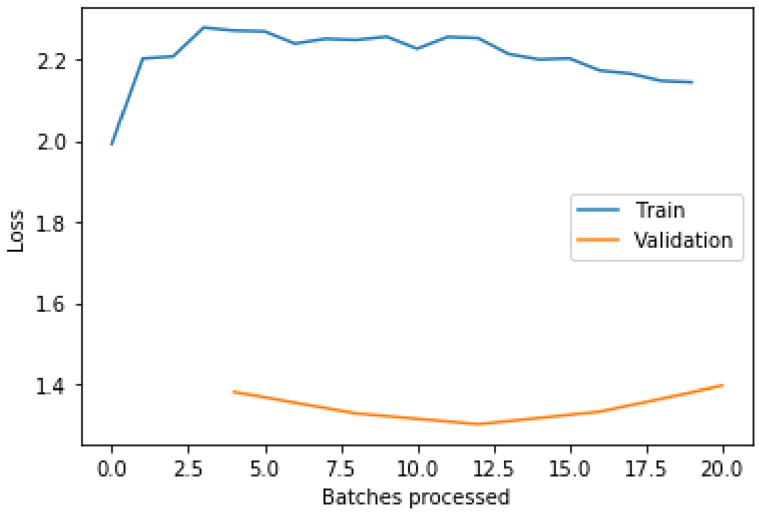
Very low learning rate

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

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

epoch	train_loss	valid_loss	error_rate	time
0	2.278331	1.381353	0.723684	00:04
1	2.250512	1.328527	0.671053	00:04
2	2.255201	1.301795	0.552632	00:04
3	2.202166	1.332711	0.526316	00:04
4	2.143343	1.396212	0.618421	00:04

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



Less number of epochs

Saved successfully!

```
net34, metrics=error_rate, pretrained=False)
```

```
learn.fit_one_cycle(1)
```

epoch	train_loss	valid_loss	error_rate	time
0	1.972283	17.870905	0.763158	00:04

More number of epochs--> Trying to overfit the model

```
np.random.seed(42)
data = ImageDataBunch.from_folder(bears_path, train=".", valid_pct=0.9, bs=32,
        ds_tfms=get_transforms(do_flip=False, max_rotate=0, max_zoom=1, max_lighting=0, ma
                             ),size=224, num_workers=4).normalize(imagenet_stats)
```

```
learn = cnn_learner(data, models.resnet50, metrics=error_rate, ps=0, wd=0)
learn.unfreeze()
learn.fit_one_cycle(40, slice(1e-6,1e-4))
```

Saved successfully!

epoch	train_loss	valid_loss	error_rate	time
0	2.328453	2.492191	0.859649	00:05
1	2.206784	2.403467	0.856725	00:02
2	2.156958	2.336805	0.859649	00:03
3	2.133642	2.250478	0.842105	00:03
4	2.042041	2.134136	0.827485	00:03
5	1.943334	1.961370	0.809942	00:03
6	1.808021	1.760963	0.742690	00:02
7	1.674848	1.557119	0.654971	00:02
8	1.546020	1.349354	0.602339	00:03
9	1.408975	1.164804	0.532164	00:03
10	1.291476	0.974762	0.415205	00:03
11	1.182727	0.840008	0.350877	00:03
12	1.088932	0.729649	0.280702	00:02
13	1.005233	0.640654	0.236842	00:02
14	0.932230	0.566220	0.204678	00:02
15	0.867518	0.509279	0.172515	00:03
16	0.809321	0.471444	0.169591	00:02
17	0.758297	0.439450	0.166667	00:02
18	0.711902	0.411721	0.152047	00:02
19	0.669631	0.388805	0.137427	00:03
[illegible]	[illegible]	[illegible]	0.125731	00:02
21	0.598300	0.366182	0.116959	00:03
22	0.566544	0.351482	0.111111	00:02
23	0.537396	0.342276	0.105263	00:02
24	0.510675	0.332557	0.096491	00:02
25	0.490786	0.324069	0.090643	00:02
26	0.467693	0.316521	0.093567	00:02
27	0.446437	0.313021	0.090643	00:02
28	0.426533	0.305634	0.096491	00:02
29	0.409575	0.296250	0.093567	00:02
30	0.392232	0.300091	0.096491	00:02
31	0.375983	0.303460	0.096491	00:02
32	0.360858	0.298624	0.096491	00:02

Saved successfully! ×