Head pose dataset

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

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

```
path = untar_data(URLs.BIWI_HEAD_POSE)
```

```
Downloading https://s3.amazonaws.com/fast-ai-imagelocal/biwi_head_pose
```

```
cal = np.genfromtxt(path/'01'/'rgb.cal', skip_footer=6); cal
```

```
array([[517.679,   0.   , 320.   ],
       [  0.   , 517.679, 240.5  ],
       [  0.   ,   0.   ,   1.   ]])
```

```
fname = '09/frame_00667_rgb.jpg'
```

```
def img2txt_name(f): return path/f'{str(f)[:-7]}pose.txt'
```

```
img = open_image(path/fname)
img.show()
```

```
ctr = np.genfromtxt(img2txt_name(fname), skip_header=3); ctr
```

```
array([187.332 ,  40.3892, 893.135 ])
```

```
def convert_biwi(coords):
    c1 = coords[0] * cal[0][0]/coords[2] + cal[0][2]
    c2 = coords[1] * cal[1][1]/coords[2] + cal[1][2]
    return tensor([c2,c1])

def get_ctr(f):
    ctr = np.genfromtxt(img2txt_name(f), skip_header=3)
    return convert_biwi(ctr)

def get_ip(img,pts): return ImagePoints(FlowField(img.size, pts), scale=True)
```

```
get_ctr(fname)
```

```
tensor([263.9104, 428.5814])
```

```
ctr = get_ctr(fname)
img.show(y=get_ip(img, ctr), figsize=(6, 6))
```

Modelling

```
data = (PointsItemList.from_folder(path)
        .split_by_valid_func(lambda o: o.parent.name=='13')
        .label_from_func(get_ctr)
        .transform(get_transforms(), tfm_y=True, size=(120,160))
        .databunch().normalize(imagenet_stats)
       )
```

```
data.show_batch(3, figsize=(9,6))
```

```
learn = cnn_learner(data, models.resnet34)
```

```
Downloading: "https://download.pytorch.org/models/resnet34-333f7ec4.pth" to /root/.ca
100%|██████████| 83.3M/83.3M [00:01<00:00, 48.0MB/s]
```

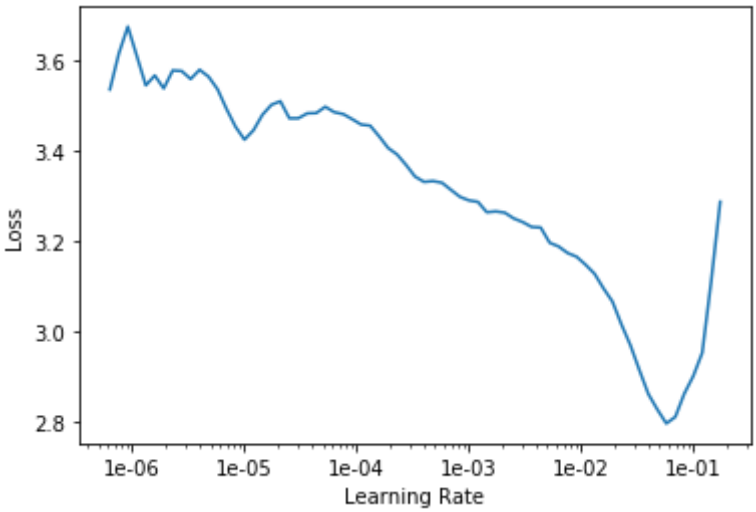
```
learn.lr_find()
learn.recorder.plot()
```

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

epoch	train_loss	valid_loss	time

35.02% [83/237 00:58<01:49 6.0438]

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




```
lr = 2e-2
```

```
learn.fit_one_cycle(5, slice(lr))
```

epoch	train_loss	valid_loss	time
0	0.109091	0.012218	02:51
1	0.039242	0.008753	02:53
2	0.013723	0.011129	02:52
3	0.008186	0.002271	02:54
4	0.006053	0.002839	02:53

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

```
learn.load('stage-1');
```

```
learn.show_results()
```

Ground truth/Predictions

additional data augmentation

```
tfms = get_transforms(max_rotate=20, max_zoom=1.5, max_lighting=0.5, max_warp=0.4, p_affin

data = (PointsItemList.from_folder(path)
        .split_by_valid_func(lambda o: o.parent.name=='13')
        .label_from_func(get_ctr)
        .transform(tfms, tfm_y=True, size=(120,160))
        .databunch().normalize(imagenet_stats)
       )

def _plot(i,j,ax):
    x,y = data.train_ds[0]
    x.show(ax, y=y)

plot_multi(_plot, 3, 3, figsize=(8,6))
```