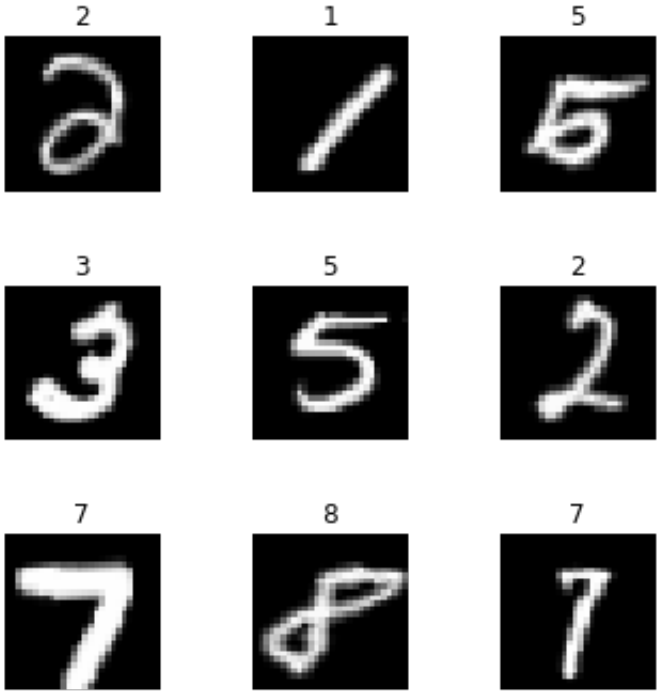
In [7]:
```
%reload_ext autoreload
%autoreload 2
%matplotlib inline

from fastai.vision import *
from fastai.metrics import error_rate
import pandas as pd
```

In [52]:
```
# learn = create_cnn(data, models.resnet34, metrics=error_rate)
help(create_cnn)

path=untar_data(URLs.MNIST)
```

In [55]:
```
data = ImageDataBunch.from_folder(path, train="training", valid="testi
ng", ds_tfms=get_transforms(do_flip=False), size=28)
data.show_batch(rows=3, figsize=(5,5))
```

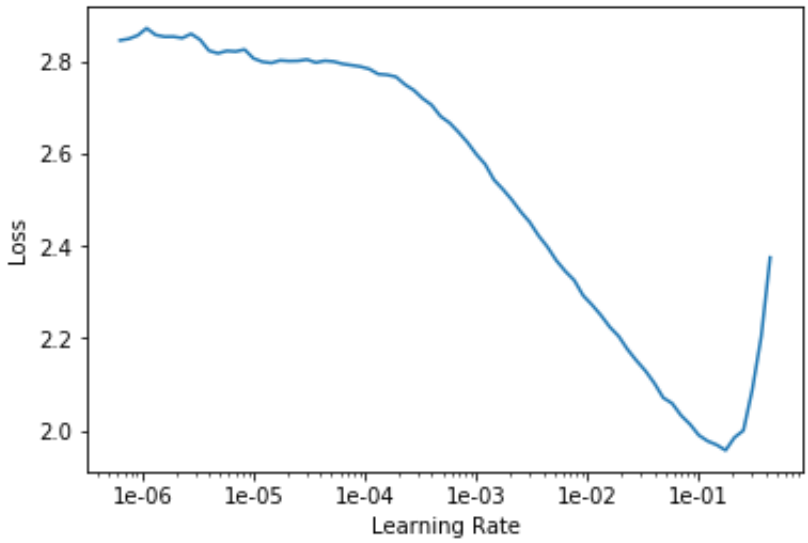


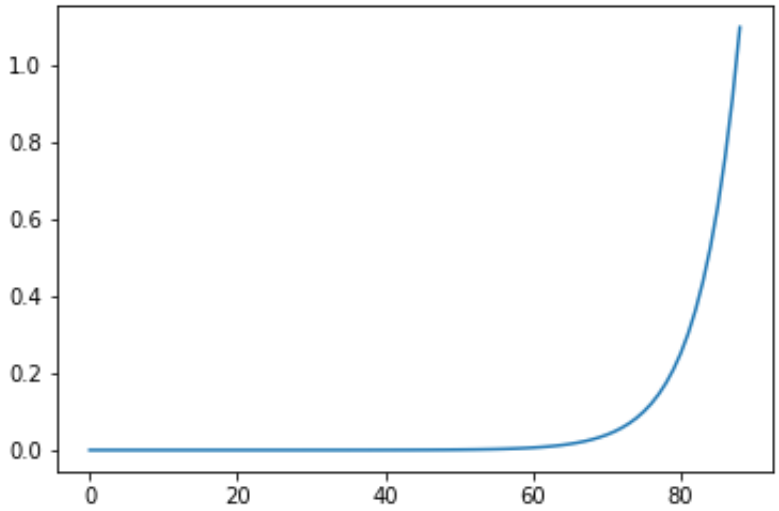
In [76]:
```
learn = create_cnn(data, models.resnet34, metrics=error_rate, pretrain
ed=False)
```

In [77]:
```
learn.lr_find()
```

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

In [78]:
```
learn.recorder.plot()

plt.figure()
learn.recorder.plot_lr()
```





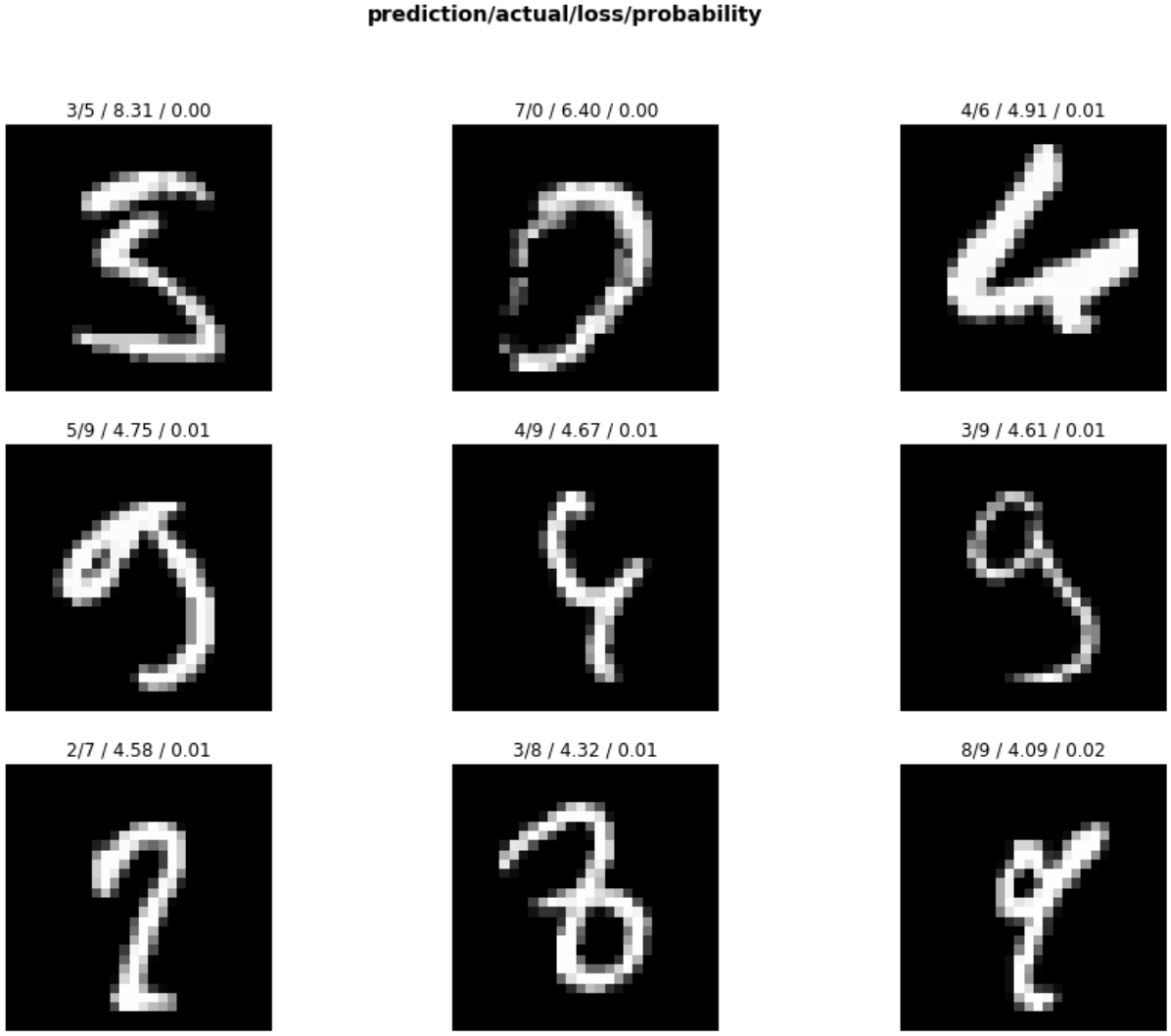
In [80]:
```
learn.fit_one_cycle(4)
```

Total time: 03:03

epoch	train_loss	valid_loss	error_rate
1	0.263670	0.155051	0.044000
2	0.119142	0.084666	0.023100
3	0.061374	0.027216	0.008600
4	0.037989	0.019857	0.006200

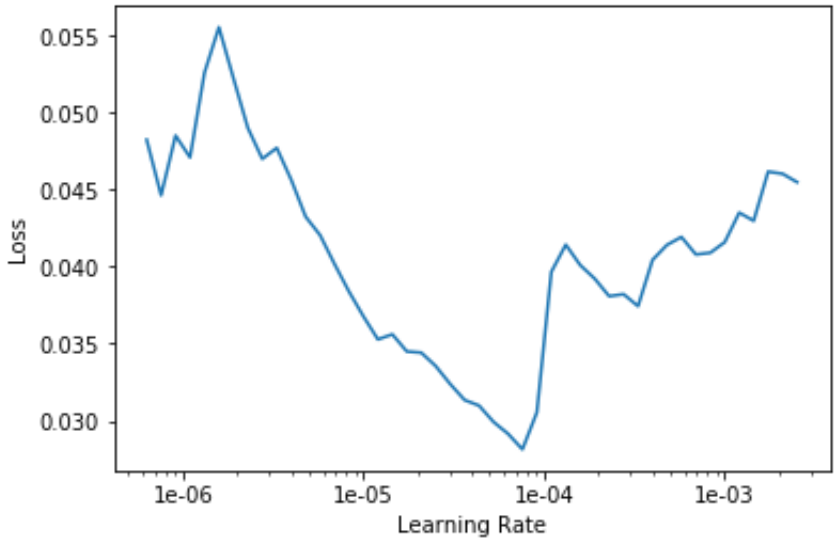
In [81]:
```
learn.save('stage-1')
```

In [82]:
```
interp = ClassificationInterpretation.from_learner(learn)

losses,idxs = interp.top_losses()

interp.plot_top_losses(9, figsize=(15,11))
```



In [83]:
```
learn.unfreeze()
learn.fit_one_cycle(1)
learn.load('stage-1')
learn.lr_find()
learn.recorder.plot()
```

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



In [84]:
```
# Final time fit based on learning rate slice

learn.unfreeze()
learn.fit_one_cycle(2, max_lr=slice(1e-5,1e-4))
```

Total time: 01:29

epoch	train_loss	valid_loss	error_rate
1	0.033732	0.021069	0.005600
2	0.024694	0.019311	0.006100