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

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

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

```
Downloading https://s3.amazonaws.com/fast-ai-imageclas/mnist_png
```

```
path.ls()
```

```
[PosixPath('/root/.fastai/data/mnist_png/testing'),
 PosixPath('/root/.fastai/data/mnist_png/training')]
```

```
il = ImageList.from_folder(path, convert_mode='L')
```

```
il.items[0]
```

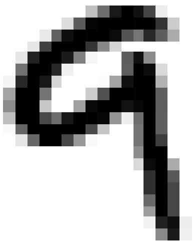
```
PosixPath('/root/.fastai/data/mnist_png/testing/9/3147.png')
```

```
defaults.cmap='binary'
```

```
il
```

```
ImageList (70000 items)
Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28,
Path: /root/.fastai/data/mnist_png
```

```
il[0].show()
```



```
sd = il.split_by_folder(train='training', valid='testing')
```

```
sd
```

```
ItemLists;

Train: ImageList (60000 items)
Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28,
Path: /root/.fastai/data/mnist_png;

Valid: ImageList (10000 items)
Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28,
Path: /root/.fastai/data/mnist_png;

Test: None
```

```
(path/'training').ls()
```

```
[PosixPath('/root/.fastai/data/mnist_png/training/9'),
 PosixPath('/root/.fastai/data/mnist_png/training/8'),
 PosixPath('/root/.fastai/data/mnist_png/training/5'),
 PosixPath('/root/.fastai/data/mnist_png/training/7'),
 PosixPath('/root/.fastai/data/mnist_png/training/4'),
 PosixPath('/root/.fastai/data/mnist_png/training/2'),
 PosixPath('/root/.fastai/data/mnist_png/training/0'),
 PosixPath('/root/.fastai/data/mnist_png/training/6'),
 PosixPath('/root/.fastai/data/mnist_png/training/3'),
 PosixPath('/root/.fastai/data/mnist_png/training/1')]
```

```
ll = sd.label_from_folder()
```

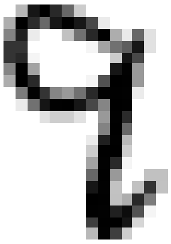
```
ll
```

```
LabelLists;

Train: LabelList (60000 items)
x: ImageList
Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28,
y: CategoryList
9,9,9,9,9
Path: /root/.fastai/data/mnist_png;

Valid: LabelList (10000 items)
x: ImageList
Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28, 28),Image (1, 28,
y: CategoryList
9,9,9,9,9
Path: /root/.fastai/data/mnist_png;

Test: None
```

```
x,y = ll.train[0]
```

```
x.show()
print(y,x.shape)
```

```
9 torch.Size([1, 28, 28])
```



```
tfms = ([*rand_pad(padding=3, size=28, mode='zeros')], [])
```

```
ll = ll.transform(tfms)
```

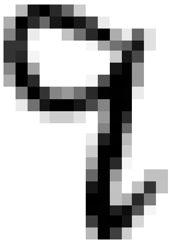
```
bs = 128
```

```
data = ll.databunch(bs=bs).normalize()
```

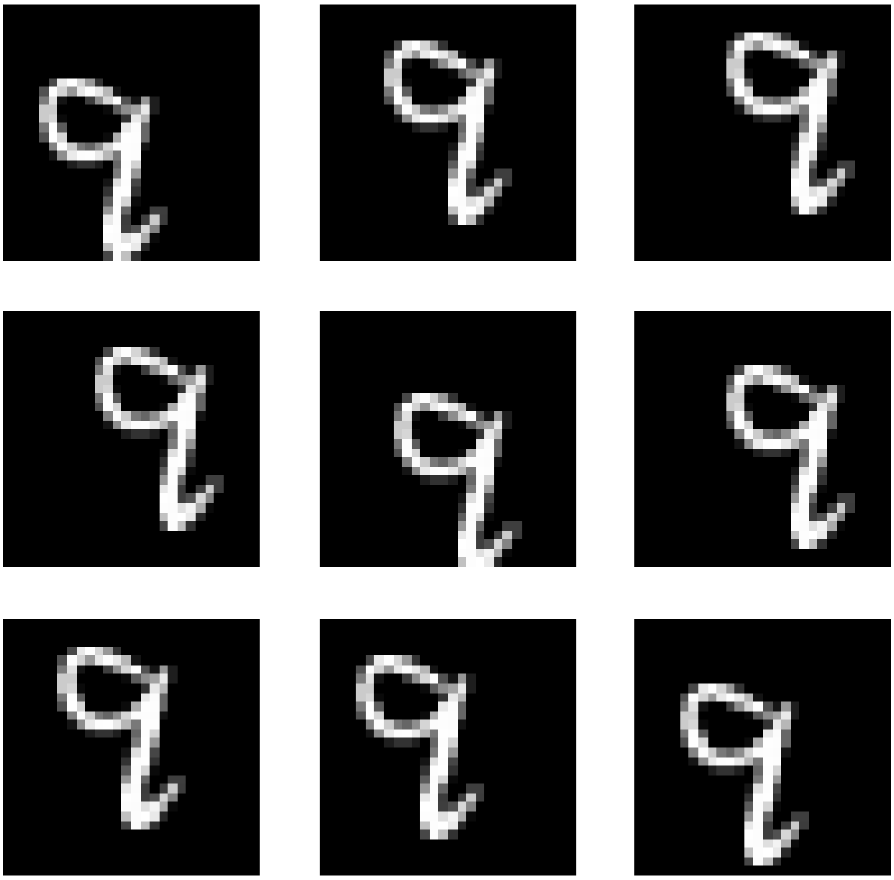
```
x,y = data.train_ds[0]
```

```
x.show()
print(y)
```

```
9
```



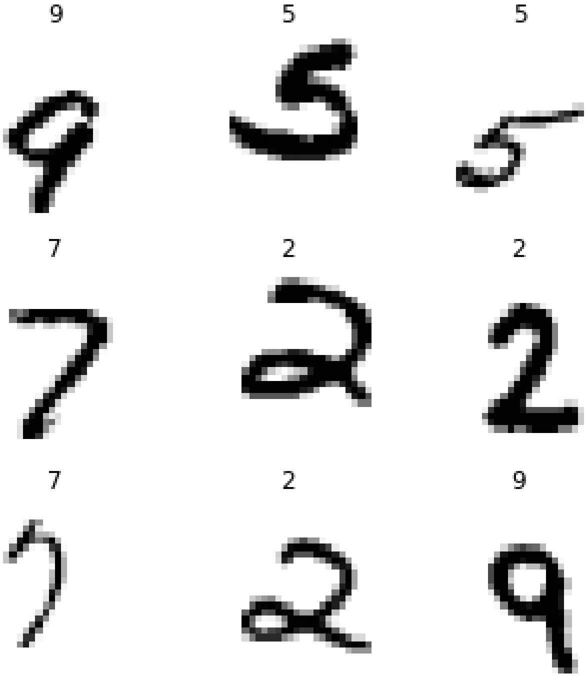
```
def _plot(i,j,ax): data.train_ds[0][0].show(ax, cmap='gray')
plot_multi(_plot, 3, 3, figsize=(8,8))
```



```
xb,yb = data.one_batch()
xb.shape,yb.shape
```

```
(torch.Size([128, 1, 28, 28]), torch.Size([128]))
```

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



```
def conv(ni,nf): return nn.Conv2d(ni, nf, kernel_size=3, stride=2, padding=1)



model = nn.Sequential(
    conv(1, 8), # 14
    nn.BatchNorm2d(8),
    nn.ReLU(),
    conv(8, 16), # 7
    nn.BatchNorm2d(16),
    nn.ReLU(),
    conv(16, 32), # 4
    nn.BatchNorm2d(32),
    nn.ReLU(),
    conv(32, 16), # 2
    nn.BatchNorm2d(16),
    nn.ReLU(),
    conv(16, 10), # 1
    nn.BatchNorm2d(10),
    Flatten()     # remove (1,1) grid
)

learn = Learner(data, model, loss_func = nn.CrossEntropyLoss(), metrics=accuracy)


print(learn.summary())
```

```
Sequential
======================================================================
Layer (type)         Output Shape         Param #    Trainable 
======================================================================
Conv2d               [8, 14, 14]          80         True      
______________________________________________________________________
BatchNorm2d          [8, 14, 14]          16         True      
______________________________________________________________________
ReLU                 [8, 14, 14]          0          False     
______________________________________________________________________
Conv2d               [16, 7, 7]           1,168      True      
______________________________________________________________________
BatchNorm2d          [16, 7, 7]           32         True      
______________________________________________________________________
ReLU                 [16, 7, 7]           0          False     
______________________________________________________________________
Conv2d               [32, 4, 4]           4,640      True      
______________________________________________________________________
BatchNorm2d          [32, 4, 4]           64         True      
______________________________________________________________________
ReLU                 [32, 4, 4]           0          False     
______________________________________________________________________
Conv2d               [16, 2, 2]           4,624      True      
______________________________________________________________________
BatchNorm2d          [16, 2, 2]           32         True      
______________________________________________________________________
ReLU                 [16, 2, 2]           0          False     
______________________________________________________________________
Conv2d               [10, 1, 1]           1,450      True      
______________________________________________________________________
BatchNorm2d          [10, 1, 1]           20         True      
______________________________________________________________________
Flatten              [10]                 0          False     
______________________________________________________________________

Total params: 12,126
Total trainable params: 12,126
Total non-trainable params: 0
Optimized with 'torch.optim.adam.Adam', betas=(0.9, 0.99)
Using true weight decay as discussed in https://www.fast.ai/2018/07/02/adam-weight-de
Loss function : CrossEntropyLoss
======================================================================
Callbacks functions applied
```

```
xb = xb.cuda()
```

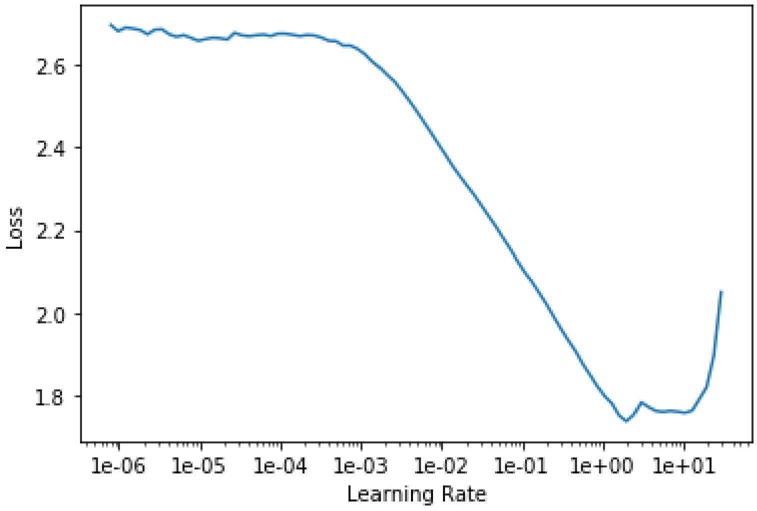
```
model(xb).shape
```

```
torch.Size([128, 10])
```

```
learn.lr_find(end_lr=100)
```

```
0.00% [0/1 00:00<00:00]
epoch  train_loss  valid_loss  accuracy  time

20.94% [98/468 00:06<00:22 2.7400]
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```

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



```
learn.fit_one_cycle(3, max_lr=0.1)
```

```
epoch  train_loss  valid_loss  accuracy  time
    0    0.219729    0.325497  0.895700  00:29
    1    0.134001    0.128912  0.958100  00:29
    2    0.072692    0.040534  0.987900  00:28
```

```
def conv2(ni,nf): return conv_layer(ni,nf,stride=2)
```

```
model = nn.Sequential(
    conv2(1, 8),   # 14
    conv2(8, 16),  # 7
    conv2(16, 32), # 4
    conv2(32, 16), # 2
    conv2(16, 10), # 1
    Flatten()      # remove (1,1) grid
)
```

```
learn = Learner(data, model, loss_func = nn.CrossEntropyLoss(), metrics=accuracy)
```

```
learn.fit_one_cycle(10, max_lr=0.1)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.243546	0.166543	0.946300	00:28
1	0.201414	0.137885	0.954900	00:28
2	0.161077	0.124679	0.960000	00:28
3	0.136109	0.157338	0.949900	00:28
4	0.118295	0.075386	0.976300	00:28
5	0.101867	0.057342	0.981100	00:28
6	0.078533	0.054512	0.981700	00:28
7	0.062270	0.041373	0.986600	00:28
8	0.049583	0.028934	0.990300	00:28
9	0.045742	0.028024	0.990500	00:29

```
class ResBlock(nn.Module):
    def __init__(self, nf):
        super().__init__()
        self.conv1 = conv_layer(nf,nf)
        self.conv2 = conv_layer(nf,nf)

    def forward(self, x): return x + self.conv2(self.conv1(x))
```

```
model = nn.Sequential(
    conv2(1, 8),
    res_block(8),
    conv2(8, 16),
    res_block(16),
    conv2(16, 32),
    res_block(32),
    conv2(32, 16),
    res_block(16),
    conv2(16, 10),
    Flatten()
)
```

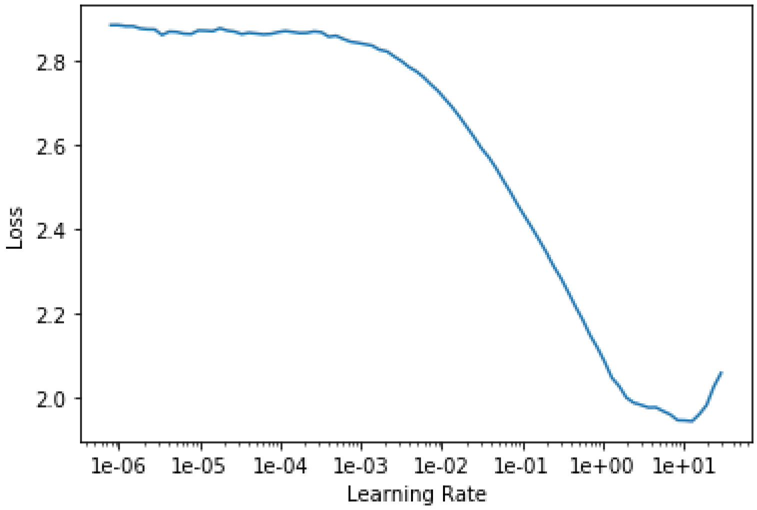
```
def conv_and_res(ni,nf): return nn.Sequential(conv2(ni, nf), res_block(nf))
```

```
model = nn.Sequential(
    conv_and_res(1, 8),
    conv_and_res(8, 16),
    conv_and_res(16, 32),
    conv_and_res(32, 16),
    conv2(16, 10),
    Flatten()
)
```

```
learn = Learner(data, model, loss_func = nn.CrossEntropyLoss(), metrics=accuracy)
```

```
learn.lr_find(end_lr=100)
learn.recorder.plot()
```

```
0.00% [0/1 00:00<00:00]
epoch  train_loss  valid_loss  accuracy  time
20.73% [97/468 00:06<00:24 2.3891]
LR Finder is complete, type {learner_name}.recorder.plot() to see the graph.
```



```
learn.fit_one_cycle(12, max_lr=0.05)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.205443	0.196096	0.949200	00:32
1	0.130249	0.482957	0.854900	00:31
2	0.108145	0.099054	0.969700	00:31
3	0.088839	0.121595	0.963300	00:31
4	0.072301	0.097752	0.972100	00:31
5	0.064870	0.037150	0.987300	00:31
6	0.047403	0.071530	0.976000	00:32
7	0.044910	0.033436	0.988800	00:32
8	0.032187	0.024436	0.992600	00:31
9	0.023785	0.017736	0.993900	00:32
10	0.021897	0.014817	0.995700	00:32
11	0.018386	0.014107	0.995900	00:32

```
print(learn.summary())
```

```
Sequential
======================================================================
Layer (type)         Output Shape         Param #    Trainable 
======================================================================
Conv2d               [8, 14, 14]          72         True      
______________________________________________________________________
ReLU                 [8, 14, 14]          0          False     
______________________________________________________________________
BatchNorm2d          [8, 14, 14]          16         True      
______________________________________________________________________
Conv2d               [8, 14, 14]          576        True      
______________________________________________________________________
ReLU                 [8, 14, 14]          0          False     
______________________________________________________________________
BatchNorm2d          [8, 14, 14]          16         True      
______________________________________________________________________
Conv2d               [8, 14, 14]          576        True      
______________________________________________________________________
ReLU                 [8, 14, 14]          0          False     
______________________________________________________________________
BatchNorm2d          [8, 14, 14]          16         True      
______________________________________________________________________
MergeLayer           [8, 14, 14]          0          False     
______________________________________________________________________
Conv2d               [16, 7, 7]           1,152      True      
______________________________________________________________________
ReLU                 [16, 7, 7]           0          False     
______________________________________________________________________
BatchNorm2d          [16, 7, 7]           32         True      
______________________________________________________________________
Conv2d               [16, 7, 7]           2,304      True      
______________________________________________________________________
ReLU                 [16, 7, 7]           0          False     
______________________________________________________________________
BatchNorm2d          [16, 7, 7]           32         True      
______________________________________________________________________
Conv2d               [16, 7, 7]           2,304      True      
______________________________________________________________________
ReLU                 [16, 7, 7]           0          False     
______________________________________________________________________
BatchNorm2d          [16, 7, 7]           32         True      
______________________________________________________________________
MergeLayer           [16, 7, 7]           0          False     
______________________________________________________________________
Conv2d               [32, 4, 4]           4,608      True      
______________________________________________________________________
ReLU                 [32, 4, 4]           0          False     
______________________________________________________________________
BatchNorm2d          [32, 4, 4]           64         True      
______________________________________________________________________
Conv2d               [32, 4, 4]           9,216      True      
______________________________________________________________________
ReLU                 [32, 4, 4]           0          False     
______________________________________________________________________
BatchNorm2d          [32, 4, 4]           64         True      
______________________________________________________________________
Conv2d               [32, 4, 4]           9,216      True      
______________________________________________________________________
ReLU                 [32, 4, 4]           0          False     
______________________________________________________________________
BatchNorm2d          [32, 4, 4]           64         True      
```

```
________________________________________________________________________
MergeLayer           [32, 4, 4]           0          False
________________________________________________________________________
Conv2d               [16, 2, 2]           4,608      True
________________________________________________________________________
ReLU                 [16, 2, 2]           0          False
________________________________________________________________________
BatchNorm2d          [16, 2, 2]           32         True
________________________________________________________________________
Conv2d               [16, 2, 2]           2,304      True
________________________________________________________________________
ReLU                 [16, 2, 2]           0          False
________________________________________________________________________
BatchNorm2d          [16, 2, 2]           32         True
________________________________________________________________________
Conv2d               [16, 2, 2]           2,304      True
________________________________________________________________________
ReLU                 [16, 2, 2]           0          False
________________________________________________________________________
BatchNorm2d          [16, 2, 2]           32         True
________________________________________________________________________
MergeLayer           [16, 2, 2]           0          False
________________________________________________________________________
Conv2d               [10, 1, 1]           1,440      True
________________________________________________________________________
ReLU                 [10, 1, 1]           0          False
________________________________________________________________________
BatchNorm2d          [10, 1, 1]           20         True
________________________________________________________________________
Flatten              [10]                 0          False
________________________________________________________________________

Total params: 41,132
Total trainable params: 41,132
Total non-trainable params: 0
Optimized with 'torch.optim.adam.Adam', betas=(0.9, 0.99)
Using true weight decay as discussed in https://www.fast.ai/2018/07/02/adam-weight-de
Loss function : CrossEntropyLoss
========================================================================
Callbacks functions applied
```