

sasank_flowers_kaggle

July 19, 2018

```
In [1]: # Put these at the top of every notebook, to get automatic reloading and inline plotting
%reload_ext autoreload
%autoreload 2
```

```
%matplotlib inline
```

```
In [2]: from fastai import *
from fastai.transforms import *
from fastai.conv_learner import *
from fastai.model import *
from fastai.dataset import *
from fastai.sgdr import *
from fastai.plots import *
```

```
In [11]: PATH = "/home/meanmachine/PycharmProjects/fastai/data/flowers/"
sz=224
```

```
In [8]: torch.cuda.is_available()
```

```
Out[8]: True
```

```
In [9]: torch.backends.cudnn.enabled
```

```
Out[9]: True
```

```
In [13]: arch = resnet34
data = ImageClassifierData.from_paths(PATH, tfms=tfms_from_model(arch, sz), test_name=)
```

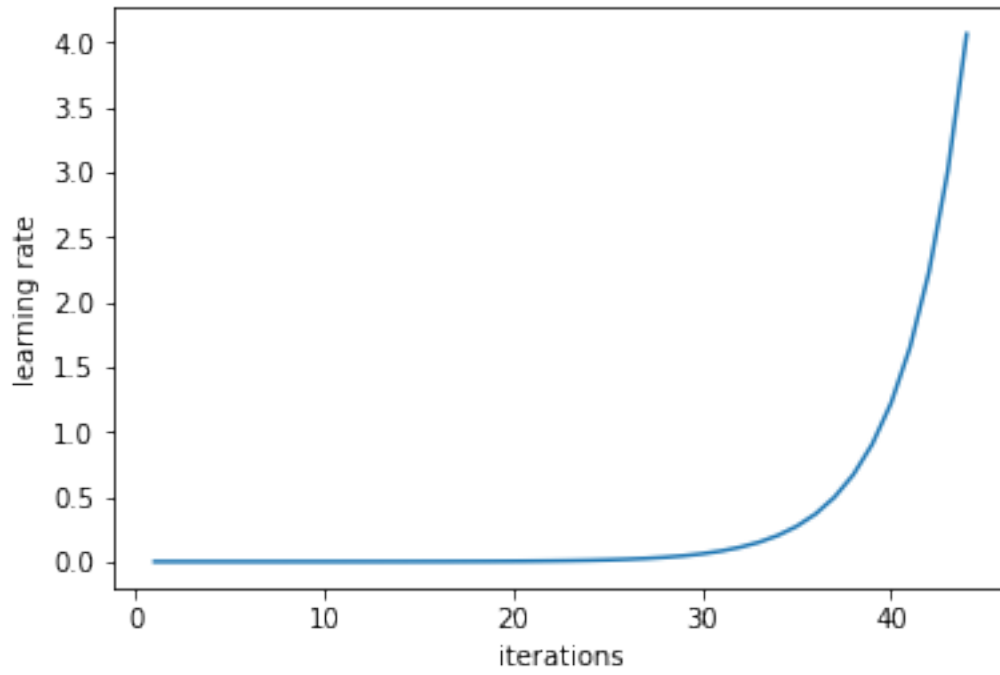
```
In [14]: learn = ConvLearner.pretrained(arch, data, precompute=True)
lrf=learn.lr_find()
```

```
100%| 1/1 [00:00<00:00, 12.26it/s]
```

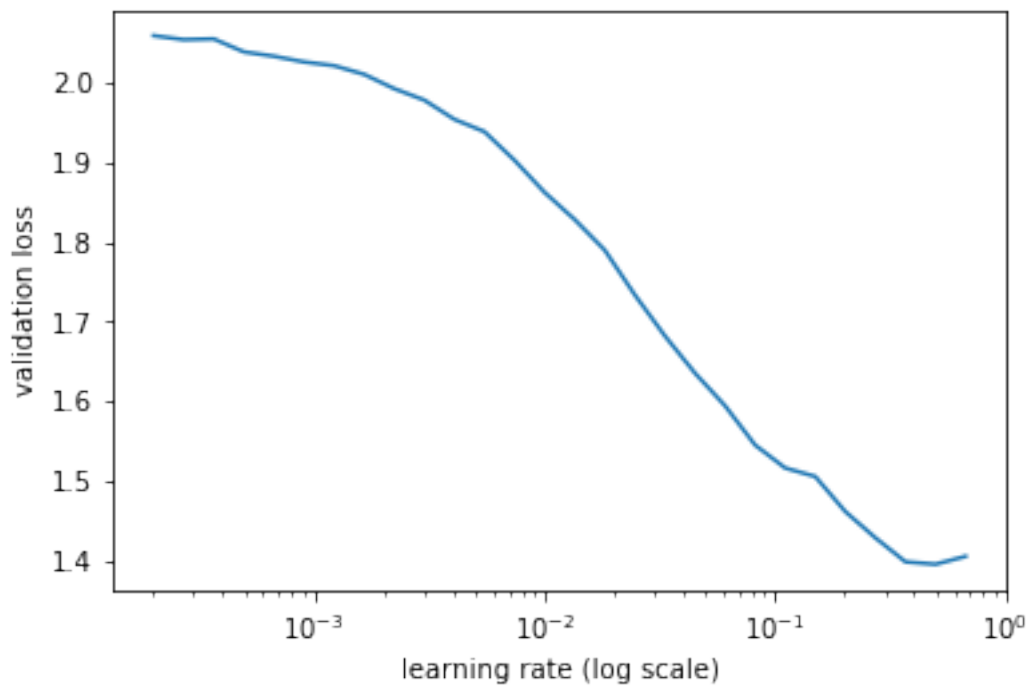
```
HBox(children=(IntProgress(value=0, description='Epoch', max=1), HTML(value='')))
```

```
41%| | 19/46 [00:00<00:00, 90.69it/s, loss=1.99]
```

```
In [15]: learn.sched.plot_lr()
```



```
In [16]: learn.sched.plot()
```



```
In [17]: data = ImageClassifierData.from_paths(PATH, tfms=tfms_from_model(arch, sz))
learn = ConvLearner.pretrained(arch, data, precompute=True)
learn.fit(1e-1, 2)
```

```
HBox(children=(IntProgress(value=0, description='Epoch', max=2), HTML(value='')))
```

| epoch | trn_loss | val_loss | accuracy |
|-------|----------|----------|----------|
| 0 | 0.876765 | 0.45727 | 0.885333 |
| 1 | 0.680776 | 0.444654 | 0.892 |

```
Out[17]: [0.4446543811162313, 0.8919999992052714]
```

```
In [18]: bs = 64 #batch size
tfms = tfms_from_model(arch, sz, aug_tfms=transforms_side_on, max_zoom=1.1)
data = ImageClassifierData.from_paths(PATH, tfms=tfms, bs=bs, num_workers=4)
learn = ConvLearner.pretrained(arch, data, precompute=True, ps=0.5)
```

```
In [19]: learn.fit(1e-1, 1)
learn.precompute=False
```

```
HBox(children=(IntProgress(value=0, description='Epoch', max=1), HTML(value='')))
```

| epoch | trn_loss | val_loss | accuracy |
|-------|----------|----------|----------|
| 0 | 0.977309 | 0.607179 | 0.832 |

```
In [20]: learn.unfreeze()
lr=np.array([1e-3,1e-2,1e-1])
```

```
In [21]: learn.fit(lr, 3, cycle_len=1)
```

```
HBox(children=(IntProgress(value=0, description='Epoch', max=3), HTML(value='')))
```

| epoch | trn_loss | val_loss | accuracy |
|-------|----------|----------|----------|
| 0 | 0.604349 | 0.337058 | 0.888 |
| 1 | 0.384032 | 0.217698 | 0.925333 |
| 2 | 0.267989 | 0.214594 | 0.937333 |

```
Out[21]: [0.21459434866905214, 0.9373333346048991]
```

```
In [22]: learn.save('flower')
```

```
In [23]: learn.load('flower')
```

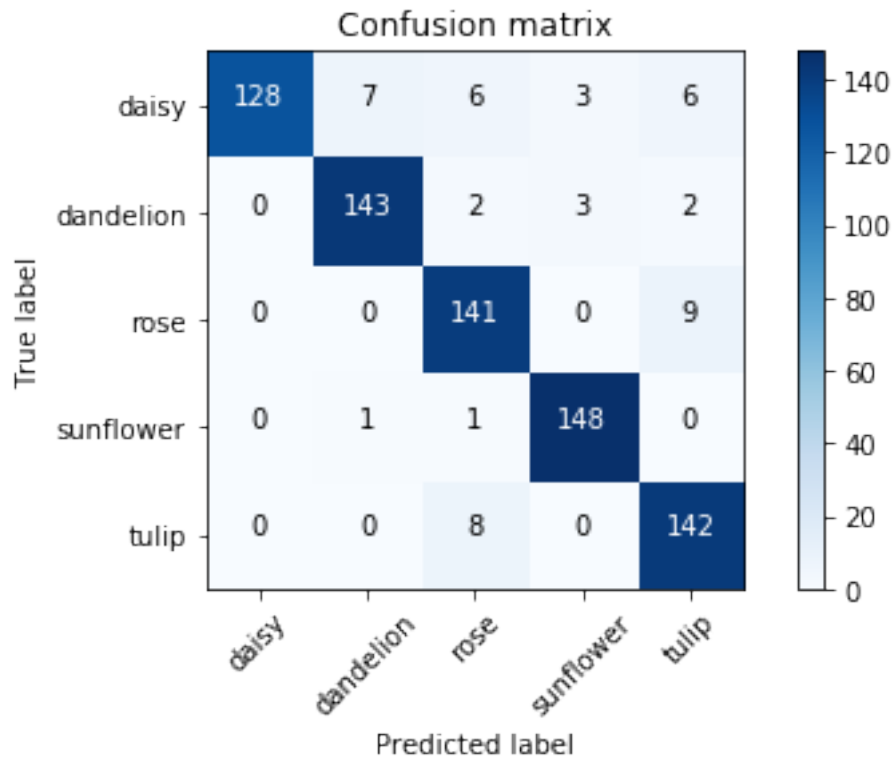
```
In [24]: log_preds,y = learn.TTA()  
         probs = np.mean(np.exp(log_preds),0)  
  
         accuracy_np(probs,y)
```

```
Out[24]: 0.936
```

```
In [25]: preds = np.argmax(probs, axis=1)  
         probs = probs[:,1]  
  
         from sklearn.metrics import confusion_matrix  
         cm = confusion_matrix(y, preds)
```

```
In [26]: plot_confusion_matrix(cm, data.classes)
```

```
[[128  7  6  3  6]  
 [  0 143  2  3  2]  
 [  0  0 141  0  9]  
 [  0  1  1 148  0]  
 [  0  0  8  0 142]]
```



```
In [27]: fn = '/home/meanmachine/PycharmProjects/fastai/data/flowers/test1/0002.jpg'
learn.precompute = False
Image.open(fn).resize((224,224))
```

Out[27]:



```
In [32]: learn.predict(is_test=True)
```

```
-----
TypeError                                 Traceback (most recent call last)

<ipython-input-32-0d02944c61ce> in <module>()
----> 1 learn.predict(is_test=True)

~/PycharmProjects/fastai/courses/dl1/fastai/learner.py in predict(self, is_test, use_sw
366     dl = self.data.test_dl if is_test else self.data.val_dl
367     m = self.swa_model if use_swa else self.model
--> 368     return predict(m, dl)
369
370     def predict_with_targs(self, is_test=False, use_swa=False):
```

```
~/PycharmProjects/fastai/courses/dl1/fastai/model.py in predict(m, dl)
244
245 def predict(m, dl):
--> 246     preda,_ = predict_with_targs_(m, dl)
247     return np.concatenate(preda)
248

~/PycharmProjects/fastai/courses/dl1/fastai/model.py in predict_with_targs_(m, dl)
256     if hasattr(m, 'reset'): m.reset()
257     res = []
--> 258     for *x,y in iter(dl): res.append([get_prediction(to_np(m(*VV(x)))),to_np(y))]
259     return zip(*res)
260
```

TypeError: 'NoneType' object is not iterable