

fast.ai Baseline

```
import pandas as pd
import numpy as np
import torch
from fastai.vision import *
from sklearn.model_selection import train_test_split

df = pd.read_csv('../input/digit-recognizer/train.csv')
col = ['pixel%d'%i for i in range(784)]

class ArrayDataset(Dataset):
    def __init__(self, image, label, num_class):
        self.image = image.astype(np.float32)
        self.label = label.astype(np.int64)
        self.c = num_class
    def __len__(self):
        return len(self.image)
    def __getitem__(self, pos):
        return Image(np.stack([self.image[pos]]*3)), self.label[pos]

X_train, X_test, Y_train, Y_test = train_test_split(df[col], df['label'],
test_size=0.1)
train_ds = ArrayDataset(X_train.values.reshape((-1,28,28)), Y_train.values,
10)
test_ds = ArrayDataset(X_test.values.reshape((-1,28,28)), Y_test.values, 10)
data = DataBunch.create(train_ds, test_ds)

learner = cnn_learner(data, models.resnet18, pretrained=False,
metrics=[accuracy])
learner.fit_one_cycle(16, 1e-4)

df = pd.read_csv('../input/digit-recognizer/test.csv')
val = df.values.reshape((-1,28,28)).astype(np.float32)

res = []
for i in val:
    img = np.stack([i]*3).reshape(1,3,28,28)
    r = learner.model(torch.tensor(img).cuda()).argmax()
    res.append(r.detach().cpu().numpy())

df = pd.read_csv('../input/digit-recognizer/sample_submission.csv')
df['Label'] = res
df.to_csv('submission.csv', index=False)
```

[fit_one_cycle](#) '1' 'One Cycle Policy'

Plugin Backlinks:

From:

<http://moro.kr/> - **Various Ways**

Permanent link:

<http://moro.kr/open/fastai-baseline>

Last update: **2021/03/31 13:55**

