周报(10/21/16)

本周进度:

- 1、论文的阅读。参阅资料熟悉了解N-neighbor-coupled network, Random network, Small-world network, scale-free work network。
- 2、代码的复现进展与遇到的问题:代码大体上已经读懂,代码的框架以及库的基本功能。遇到的问题是对于启动程序的某个参数的设定不能正确的预测。分析问题:在数据的输入方面,由于Ubuntu的内存不足,将数据缩小,每个样本的宽度达不到858的像素点,然后对代码进行修改,可能引起了其它的变化。拟定采取的方法为:1)、向作者发邮件询问有关细节,暂时并没有回复。2)、阅读其它类似的CNN结合RNN的代码,以求得到解答。
- 3、有关网络结构的矩阵处理。目标是解决编号不同带来的矩阵不同的问题。利用正常的矩阵线性变化,左乘与右乘处理之后可以达到。不过我们需要思考线性变换的目标是什么,靠近主对角线还是尽量的稀疏。目标是通过一定的策略,将网络中的每一个结点进行排序。
- 4、周二晚上开始跑程序, tc_net_rnn_onernn.py文件。这是RNN与CNN结合的一个程序, 选取这个文件的原因是在论文中, 该方法的准确率最高。初始500步, 用13小时进行了60步, 速度感人。准确率方面, 训练数据的准确率达到97.83, 然而验证的准确率却上不了20.00。原因不明, 有待进一步的研究。另外程序刚刚起步的准确率低的惊人, 也不知道什么原因。

下周计划:

- 1、Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers,并且实现代码。
 - 2、对网络中结点排序的理论验证。

本周结果:

学习的成果:

```
5poken-language-identification-master - [~/下载/Spoken-language-identification-master] - .../theano/main.py - PyCharm Community Edition 4.5.5
                                                                                                                                                                                                                                                                                                                                   ■ 🚺 🕩 11:49 🖔
            File Edit View Navigate Code Refactor Run Iools VCS Window Help
Ø
            □ Spoken-language-identification-master ⟩ □ theano ⟩ ဨ main.py ⟩

# © Project ▼ © + | ® - !* | @ tc_net_mn_onernn.py × | @ plot.py × | @ augment_data.py × | @ base_network.py × | @ main.py × | @ main.py ×
                                                                                                                                                                                                                                                                                                                   augment_data ▼ ▶ ∰ Q
a,
                     □ Spoken-language-identification
                                                                                                                                             x in all_prediction:
print >> pred_csv, ",".join([("%.6f" % prob) for prob in x])
                     ▶ □ ensembling
▶ □ prototxt
▼ □ theano
                        ▼ □ networks

    init_.py
    base_network.py

                                                                                                       if args.mode == 'train':
    print "==> training"
    for epch in range(start_epoch, args.epochs):
        do_epoch('train', epoch)
        test_loss = do_epoch('test', epoch)
        state_name = '%s-epoch%d.test'.5f.state' % (network_name, epoch, test_loss)
        print "==> saving ... %s" % state_name
        network.save_params(state_name, epoch)
                                tc_net_mod.py
tc_net_mod_5khz_small.py
                                                                                                        elif args.mode == 'test':
do_epoch('predict', 0)
elif args.mode == 'test_on_train':
do_epoch('predict_on_train', 0)
else:
                                tc_net_rnn_nodense.py
                                  tc_net_rnn_onernn.py
                                letc_net_rnn_onernn_notime
letc_net_rnn_shared.py
                                 tc_net_rnn_shared_pad.py
                          cacuracy: 97.83 percent
accuracy: 11.98 percent
accuracy: 11.98 percent

=== saving ... tc_net_rnn.4conv.pad.GRU.onernn.num_units508.5khz.bs32.bn.epoch58.test3.81294.state
accuracy: 97.83 percent
=== saving ... tc_net_rnn.4conv.pad.GRU.onernn.num_units508.5khz.bs32.bn.epoch59.test3.81450.state
accuracy: 97.83 percent
=== saving ... tc_net_rnn.4conv.pad.GRU.onernn.num_units508.5khz.bs32.bn.epoch69.test3.81450.state
accuracy: 97.83 percent
==> saving ... tc_net_rnn.4conv.pad.GRU.onernn.num_units508.5khz.bs32.bn.epoch60.test3.81598.state
accuracy: 11.98 percent
==> saving ... tc_net_rnn.4conv.pad.GRU.onernn.num_units508.5khz.bs32.bn.epoch60.test3.81735.state
accuracy: 97.83 percent
==> saving ... tc_net_rnn.4conv.pad.GRU.onernn.num_units508.5khz.bs32.bn.epoch61.test3.81735.state
accuracy: 97.83 percent
                   % 6: TODO S Python Console ☑ Terminal
                                                                                                                                                                                                                                                                                                                          132:12 LF≎ UTF-8≎ 🚡 🚇
```

Figure 1: Based on Ubuntu