

# 基于 Matlab 神经网络的地震事件检测

苏文君柳，李远芳

2019 年 6 月 10 日

## 目录

<b>1</b>	<b>研究背景及意义</b>	<b>2</b>
<b>2</b>	<b>研究流程</b>	<b>2</b>
2.1	地震目录的下载 . . . . .	2
2.1.1	Anaconda 的安装 . . . . .	3
2.1.2	ObsPy 的安装 . . . . .	3
2.1.3	利用 ObsPy 下载地震目录 . . . . .	5
2.2	对地震目录结果进行聚类 . . . . .	5
2.3	地震波形的下载 . . . . .	7
2.3.1	事件波形的下载 . . . . .	7
2.3.2	噪音波形的下载 . . . . .	9
2.4	神经网络的构建 . . . . .	11
2.4.1	神经网络及其分类 . . . . .	11
2.4.2	基于 Matlab 的神经网络训练 . . . . .	11
<b>3</b>	<b>结果分析</b>	<b>15</b>
<b>4</b>	<b>总结与展望</b>	<b>19</b>
4.1	结论 . . . . .	19
4.2	研究过程发现的问题 . . . . .	19
4.3	展望与致谢 . . . . .	20

## 1 研究背景及意义

本文为理论地震学课程的结课论文，结合地震学前沿课程和自己的兴趣爱好，选择了当下最为流行的机器学习/神经网络研究地震事件的检测。2019年2月25日前后，四川自贡发生多次4-5次地震，此事件被当地人民群众认为是页岩气开采所致。此类人工诱发地震往往和天然地震不同，震源机制和震级可能都有其自身的特点。因此，本文主要利用了Thibaut Perol [?]等人于2018年的研究成果，将卷积神经网络这一机器学习方法，利用Matlab复现，并以Oklahoma州为例检测潜在的地震事件情况。

与此同时，本次研究涉及到获取地震波形、简单的数据处理工作，对今后的研究工作有一定的作用。文中所使用的语言为Matlab和Python，安装过程详细，可以作为初学者进行入门操作的手册。

## 2 研究流程

本次研究流程较为简单，主要的流程为

- 利用Obspy下载地震目录
- 利用Matlab中的irisFetch获取地震波形数据
- 利用K-means方法进行聚类，分出几类不同地点的地震波的噪声，并进行机器学习
- 利用训练好的神经网络对波形进行检测，以找出未被检测的地震事件

### 2.1 地震目录的下载

本节中我们将详述地震事件和目录的下载流程。我们将使用Obspy这一工具包进行目录的下载。

Obspy是一套基于Python地震数据处理工具包，于2010年由Beyreuther等人编撰并于Github开源，目前与2019年更新至1.1.1版本。其官方手册(<http://docs.obspy.org/>)在Github上托管。该工具包秉承Python良好的特性——使用简单函数见名知意，面向对象编程更方便了程序的编写，使得科研工作者可以专注于目的而不是程序本身。

### 2.1.1 Anaconda 的安装

安装 Obspy 时，强烈建议通过 Anaconda 进行安装。Anaconda 是一个开源的 Python 版本，包含了许多 Python 所需要的科学计算包，如 SciPy, NumPy 等。Anaconda 的官方网站为 <https://www.anaconda.com/>，可以直接在 Windows 上下载并安装。Python 的优势和劣势都是它拥有的广泛的第三方库，利用 Anaconda 安装第三方库可以自动帮助我们检测依赖环境。值得注意的是，Anaconda 的商标是被保护的，因此国内镜像源（如清华、中科大等）于 2019 年 4 月已经停止更新。安装 Anaconda 后，可以通过 cmd 或者 Windows Powershell 输入

```
1 anaconda -V
```

如果出现了 Anaconda 的版本信息，则说明 Anaconda 安装成功。

### 2.1.2 ObsPy 的安装

Obspy 的代码在 Anaconda Cloud 上托管，conda-forge 是为了维护和建立软件包所发起的。1.1.1 版本的 ObsPy 可在网站 <https://anaconda.org/conda-forge/obspy> 上找到。如果有其他基于 Python 的程序需要执行，强烈建议为不同的 Python 项目建立不同的环境，以解决各个第三方库之间的冲突。

环境的建立过程如下：

在 cmd 中输入

```
1 conda create -n obspyEnv python=3.5  
activate obspyEnv
```

第一条命令是创建名为 obspyEnv 的环境，Python 版本为 3.5。之后系统会下载一定的初始库，下载完成后利用第二条命令即可激活 obspyEnv 环境。如果激活成功，在 cmd 的用户提示符之前，会有 (obspyEnv) 的提示字样。

值得注意的是，如果编写 Python 时使用的是 VScode，则可能会出现一定的问题。原因是 VScode 内部调用的命令程序是 Windows Powershell

而不是 cmd，因此在 VScode 的命令行下由于权限的原因无法激活环境。如果遇到这样的问题可以利用

```
conda install -n root -c pscondaenvs pscondaenvs
```

安装 PSCondaEnvs 包后，执行命令

```
1 Set-ExecutionPolicy RemoteSigned
```

改变 PowerShell 的安全策略，这样就可以在 VScode 下改变环境  
创建完环境后，即可使用命令

```
1 conda install -c conda-forge obspy
```

安装 ObsPy。值得注意的是，安装 obspy 后运行程序时，可能会导致 NumPy 的冲突，使得程序无法运行，这是由于 ObsPy 和 Anaconda 都自带了 NumPy 库，如果遇到这样的问题，可以利用 pip 重新安装 NumPy 库，命令如下

```
1 pip uninstall numpy  
pip install numpy
```

Python 程序的编写建议使用 VSCode 或者 jupyter notebook 进行编写。VSCode 具有轻量化的特点，丰富的插件使得编程更为简单（可以提供括号、命令补全等功能），VSCode 可以在 <https://code.visualstudio.com/> 下载。jupyter notebook 则可以提供分段运行函数，并且其有丰富的文本支持功能，可以方便的输出 pdf 格式，将程序、程序结果和必要的文字说明统一输出，jupyter notebook 可以通过 Anaconda 进行安装和使用，命令为

```
conda install jupyter notebook
```

安装完成后在命令行直接输入 jupyter notebook 即可打开。

### 2.1.3 利用 ObsPy 下载地震目录

安装 ObsPy 后即可利用它通过 IRIS 下载地震事件目录。代码如下

```

1  from obspy.clients.fdsn import Client
    client = Client("IRIS")
3  from obspy import UTCDateTime

5  starttime = UTCDateTime("2014-02-15")
    endtime = UTCDateTime("2016-11-16")
7  #下载哪些时间段内的目录
    cat = client.get_events(starttime=starttime, endtime=endtime,
9                           minlatitude=35.70,maxlatitude=36.00,
                           minlongitude=-97.6,maxlongitude=-97.2,
11                          catalog="ISC")

13

15  f = open("cat.txt", "w")
    print(cat.__str__(print_all=True), file=f)
    #将结果输出到cat.txt, 这里一定要注意print_all=True, 输出完所有的目录

```

运行后, 其 cat.txt 文件部分显示结果如下

```

703 Event(s) in Catalog:
2  2015-08-31T01:04:03.480000Z | +35.761,  -97.371 | 3.0 mb_Lg
    2015-08-27T03:14:23.200000Z | +35.954,  -97.410 | 2.7 mb_Lg
4  2015-08-26T08:43:54.520000Z | +35.717,  -97.457 | 2.2 mb_Lg
    2015-08-26T07:56:39.510000Z | +35.722,  -97.467 | 2.3 mb_Lg
6  2015-08-25T15:07:58.680000Z | +35.891,  -97.277 | 2.5 mb_Lg
    2015-08-25T14:46:11.480000Z | +35.906,  -97.273 | 2.7 mb_Lg
8  2015-08-24T16:09:41.560000Z | +35.840,  -97.422 | 2.9 mb_Lg
    2015-08-14T10:34:39.540000Z | +35.705,  -97.495 | 2.3 mb_Lg
10 2015-08-13T14:48:09.830000Z | +35.767,  -97.405 | 2.6 mb_Lg
    2015-08-13T14:32:58.510000Z | +35.756,  -97.389 | 2.4 mb_Lg
12 2015-08-13T13:59:41.710000Z | +35.760,  -97.458 | 3.2 mb_Lg
    2015-08-12T08:54:40.840000Z | +35.750,  -97.388 | 1.9 ML

```

## 2.2 对地震目录结果进行聚类

本节中将对地震目录进行 K-means 聚类分析, 判断出几类地震。聚类顾名思义, 是定义并计算某些点/事件之间的距离后, 根据距离和给定的类

的数目进行分组，使得每一组组内的距离都最小。Matlab 自带的聚类分析函数即可将事件进行聚类。

我们需要对利用 ObsPy 得到的目录进行处理，这里可以 Matlab 自动生成的文件导入函数 `importfile`。导入后利用 `kmeans` 函数将地震事件分为三类，代码如下

```

1  %% 读取地震目录
   cat = importfile('cat.txt', 2, 731);
3  %% 这里是用python下载的

5  %% 聚类，打标签
   for ii=1:length(cat)
7      data.longitude(ii)=str2double(cat(ii,4));
      data.latitude(ii)=str2double(cat(ii,3));
9  end
   figure;
11  plot(data.longitude,data.latitude, '. ');
   title 'Oklahoma州地震事件位置';

13
   clusterEventData(:,1)=data.longitude';
15  clusterEventData(:,2)=data.latitude';
   clear data
17  NumCluster=3;
   opts = statset('Display','final');
19  [idx,C] = kmeans(clusterEventData,NumCluster,'Distance','cityblock',...
   'Replicates',5,'Options',opts);
21  %%进行聚类，

23  figure;
   legendCluster=[];
25  for ii=1:NumCluster
       plot(clusterEventData(idx==ii,1),clusterEventData(idx==ii,2),'o')
27       hold on
       str=strcat('Cluster',num2str(ii));
29       legendCluster=[legendCluster;str];
   end
31  legend(legendCluster)
   title 'Oklahoma州地震事件聚类结果'
33  hold off

```

聚类的结果如下图所示。

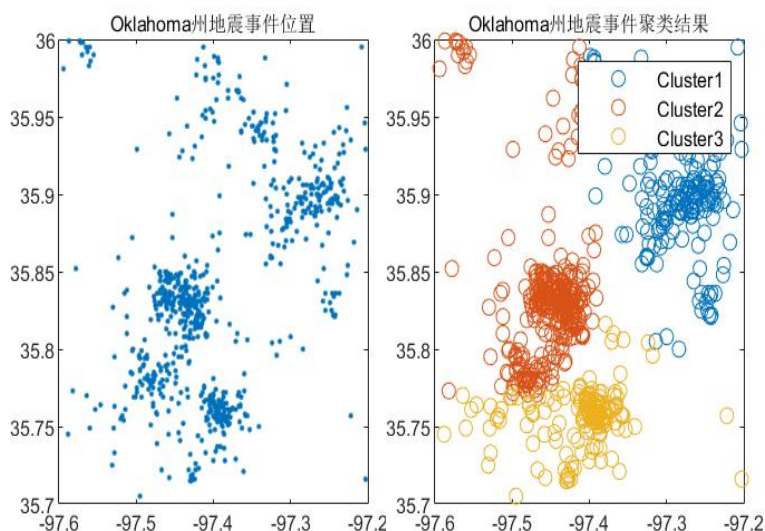


图 1: 地震事件空间分布

图 2: 地震事件聚类分析结果

## 2.3 地震波形的下载

### 2.3.1 事件波形的下载

IRIS 提供了 `irisFetch` 函数进行数据下载。我们根据之前得到的 `Cat` 目录，下载 GS.OK029 台站的数据，波形数据为发震时间后 10s，根据 GS.OK029 台站的采样率，得到的采样点数为 1000 个。将三个分量的数据拼接即可得到长度为 3000 的列向量，利用前节得到的聚类结果为这些数据打上相应的标签，即可用于后节的神经网络训练。此外，下载到的波形数据为了训练的方便，需要进行归一化，可以利用线性归一化，将数据映射到 [0,1]。Matlab 代码如下

```

1  %% 下载地震波形，并且整理数据集
3  ratioNoiseEvent=3;
   %空窗（噪声）和事件的比值
5
   NumNoise=length(cat)*ratioNoiseEvent;
7  data_cluster=zeros(NumCluster+1,length(cat)+NumNoise);
   %这里被坑了，记住要NumCluster+1，有噪声
9  %%%%%%%%%%
   startTime=datetime(strcat(cat(1,1),{32},cat(1,2)));

```

```

11     endTime=startTime+seconds(10);
12     mytrace=irisFetch.Traces('GS','OK029','00','HH*',datestr(startTime,'yyyy
    -mm-dd HH:MM:SS'),datestr(endTime,'yyyy-mm-dd HH:MM:SS'));
13     waveform=mytrace(1).data;
    %%%%这里是为了初始化，找一个长度用的。有点坑
15     data_wave=zeros(length(waveform),3,length(cat)+NumNoise);

17     h=waitbar(0,"正在下载数据制作数据集(事件部分)");
    for ii=1:length(cat)
19         startTime=datetime(strcat(cat(ii,1),{32},cat(ii,2)));
        % {32} 是一个空格
21         endTime=startTime+seconds(10);
        % 我们做10秒的波

23         mytrace=irisFetch.Traces('GS','OK029','00','*',datestr(startTime,'
            yyyy-mm-dd HH:MM:SS'),datestr(endTime,'yyyy-mm-dd HH:MM:SS'));
25         % 从左到右分别是, Net, Sta, Loc, Cha, Starttime, Endtime [,quality][,
            includePZ][,verbosity]
        %     EHZ/EHN/EHE Short Period 100 sps
27         %     BHZ/BHN/BHE Broad Band 20 sps
        %     LHZ/LHN/LHE Long Period 1 sps
29         %     VHZ/VHN/VHE Very Long Period 0.1 sps
        %     对于GS.OK029,只有HH和LH,我们用HH

31         if isempty(mytrace)==1
33             DataMiss(ii)=1;
            continue;
35         end
        % 有些数据没有? 跳过它

37         for jj=1:3
39             maxvalue=max(mytrace(jj).data);
            minvalue=min(mytrace(jj).data);
41             % data_wave(:,jj,ii)=(mytrace(jj).data(1:1000)-minvalue)/(
                maxvalue-minvalue);
            % ??? 有时会变成1001个点? 怎么回事。
43             % 线性归一化
            % 要一起归一化, 否则会有“断头台”

45         end
        % 这里做分类
47         data_cluster(idx(ii),ii)=1;
        str=['君柳哥我在拼命下载波形数据...',num2str(ii/length(cat)*100),'%']
            ];
49         waitbar(ii/length(cat),h,str)

51     end

```



```
close(h)
```

值得注意的是，下载的波形数据不一定是完整的。某些时间段没有对应的数据，因此 Trace 结构体就会为空，需要单独做判断。另外，有时候下载到的波形数据可能是 1001 个点，这时候需要截掉最后的几个点。这里的归一化出现了一定的问题，正确的做法应该是将所有的波形下载后一起归一化，这里分段归一化，导致将三个分量的数据拼接在一起的时候会出现“断头”的现象，如下图所示。

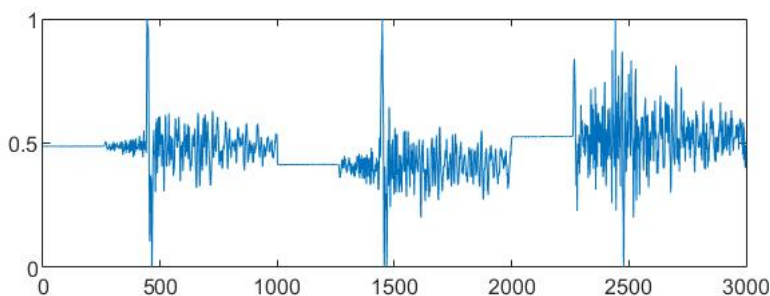


图 3: 由于没有合适归一化造成的跳跃现象

### 2.3.2 噪音波形的下载

为了使得神经网络可以识别噪音和事件，我们还需要下载噪音数据进行训练。这里的噪音选取了地震目录的间隔。例如 1 月 1 日凌晨 1 时和凌晨 2 时有两个事件，那么就取 1 时 30 分作为噪音数据。为了更好的选取这些噪音，如果有一个相对准确的地震目录，那么我们取到“准确噪音”的可能性也越大。这些噪音就打上 4 号标签（如果前节中聚类数目为 3 类），其 Matlab 代码如下。

```
h=waitbar(0,"正在下载数据制作数据集(噪音部分)");
2 figure
3 for ii=0:length(cat)-2
4     startTime=datetime(strcat(cat(ii+1,1),{32},cat(ii+1,2)));
5     nextStartTime=datetime(strcat(cat(ii+2,1),{32},cat(ii+2,2)));
6     intervalSeconds=(posixtime(startTime)-posixtime(nextStartTime))/2;
```

```

8      startTime=nextStartTime+seconds(intervalSeconds);
      %我擦咧，这个顺序是倒序的，坑啊！

10

      %噪音通过两个事件中间的地方取
12      for jj=1:ratioNoiseEvent
          endTime=startTime+seconds(10);
14          mytrace=irisFetch.Traces('GS','OK029','00','HH*',datestr(
              startTime,'yyyy-mm-dd HH:MM:SS'),datestr(endTime,'yyyy-mm-
              dd HH:MM:SS'));
          if isempty(mytrace)==1
16              DataMiss(length(cat)+ii*ratioNoiseEvent+1)=1;
              DataMiss(length(cat)+ii*ratioNoiseEvent+2)=1;
18              DataMiss(length(cat)+ii*ratioNoiseEvent+3)=1;
              continue;
20          end
          %有些数据没有？跳过它
22          for kk=1:3
              maxvalue=max(mytrace(kk).data);
24              minvalue=min(mytrace(kk).data);
              %
              data_wave(:,kk,length(cat)+ii*ratioNoiseEvent+jj)=(mytrace(
26                  kk).data(1:1000)-minvalue)/(maxvalue-minvalue);
              %这里接着原来的data_wave数据
          end
28          subplot 311
              plot(data_wave(:,kk,length(cat)+ii*ratioNoiseEvent+1));
30          subplot 312
              plot(data_wave(:,kk,length(cat)+ii*ratioNoiseEvent+2));
32          subplot 313
              plot(data_wave(:,kk,length(cat)+ii*ratioNoiseEvent+3));
34          data_cluster(NumCluster+1,length(cat)+ii*ratioNoiseEvent+jj)=1;
          %全是噪声
36          startTime=endTime;
      end
38      str=['君柳哥我在拼命下载噪声数据...',num2str(ii/length(cat)*100),'%']
          ];
          waitbar(ii/length(cat),h,str)
40
42      end
      close(h)

```

得到的噪音数据如下图所示

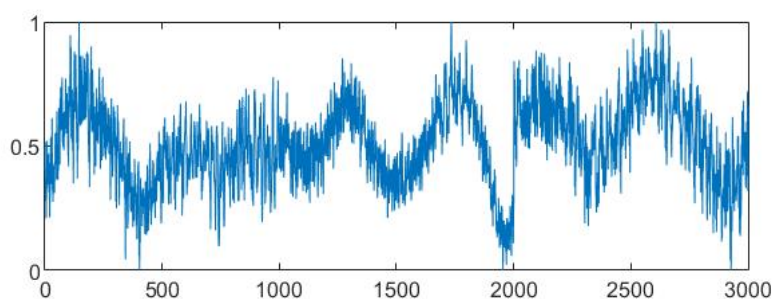


图 4: 噪音数据

## 2.4 神经网络的构建

本节中将详述基于 Matlab 的神经网络构建。

### 2.4.1 神经网络及其分类

在机器学习和相关领域，人工神经网络（人工神经网络）的计算模型灵感来自动物的中枢神经系统（尤其是脑），并且被用于估计或可以依赖于大量的输入和一般的未知近似函数。人工神经网络通常呈现为相互连接的“神经元”，它可以从输入的计算值，并且能够机器学习以及模式识别由于它们的自适应性质的系统。

神经网络有许多类型，如卷积神经网络 CNN，深度信念网络 DBN，循环神经网络 RNN 等。大部分的神经网络都是利用 Sigmoid 函数或者 softmax 等函数模拟神经元数据，并且构建目标函数，通过各种最优化方法（如共轭梯度等）求得目标函数最小值，进而训练神经网络。

### 2.4.2 基于 Matlab 的神经网络训练

Matlab 自带了神经网络工具箱，在 Matlab2018a 中自带了 Neural Pattern Recognition (nprtool) 工具箱训练神经网络。在命令行中输入 nprtool 即可打开工具箱。

根据提示信息将数据输入至工具箱并进行训练，训练过程如图所示

当训练集数目为 2595 时（事件和噪声比为 1:3），其训练的结果如下图所示。

根据结果看，虽然训练的结果较好（Training Confusion Matrix 最下面一行），但是在测试集中，第一类和第二类的测试结果准确率只有 40% 和

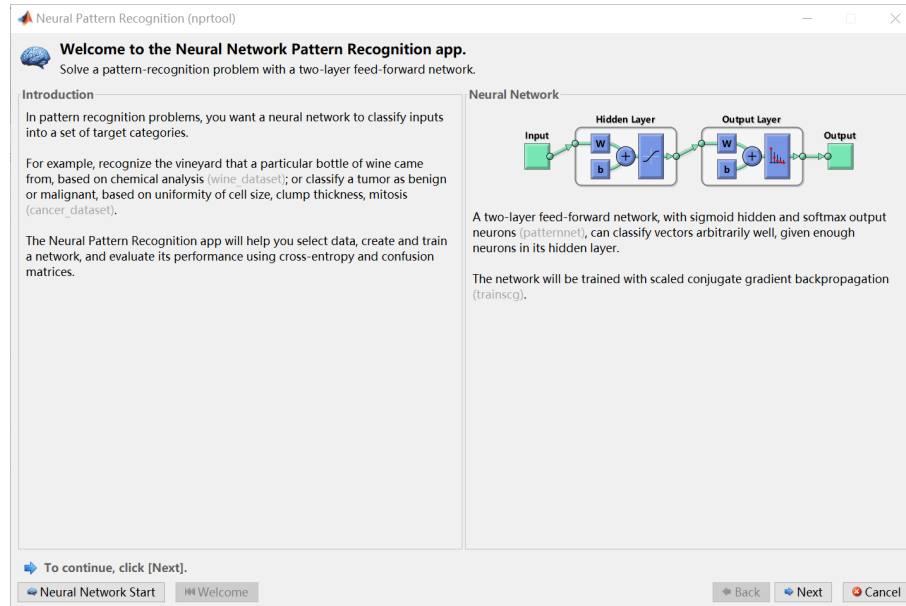


图 5: nprtool 工具箱

18.4%，甚至不如瞎猜的准确度高 (25%)，主要原因是训练集太小。为了解决训练集较小的问题，我们人为将事件数据增加最大值 2% 和 5% 的高斯白噪声，新生成两个数据集 DatasetAddNoise1 和 DatasetAddNoise2，增加噪声后的波形数据如下图所示。

我们将这两部分的噪声加入训练集合中，下图分别为加入训练集 DatasetAddNoise1 和训练集 DatasetAddNoise1 与 DatasetAddNoise2 的训练结果。

其 Matlab 代码如下

```

for i=1:length(data_cluster)
2   data_waveSys(:,i)=data_wave(:,i)+0.02*max(data_wave(:,i))*rand(3000,1);
   minvalue=min(data_waveSys(:,i));
4   maxvalue=max(data_waveSys(:,i));
   data_waveSys(:,i)=(data_waveSys(:,i)-minvalue)/(maxvalue-minvalue);
6 end

8
for i=1:length(data_cluster)
10  data_waveSys2(:,i)=data_wave(:,i)+0.05*max(data_wave(:,i))*rand(3000,1);
   minvalue=min(data_waveSys2(:,i));
12  maxvalue=max(data_waveSys2(:,i));

```

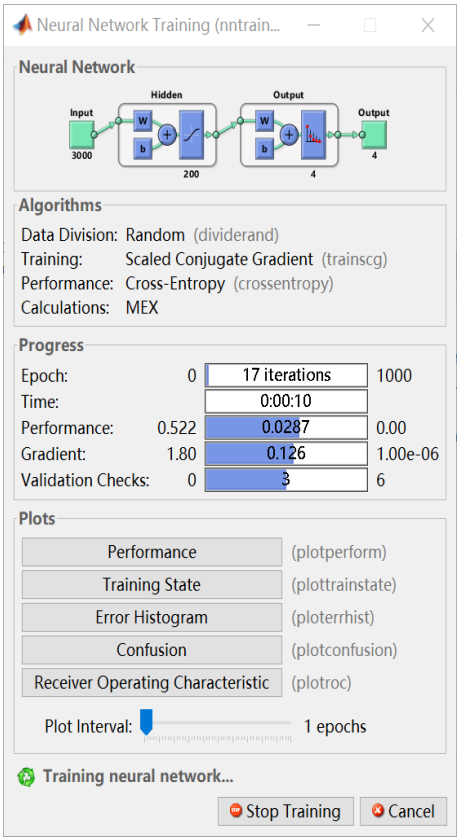


图 6: nprtool 训练过程



图 7: 训练数目为 2595 时的训练结果

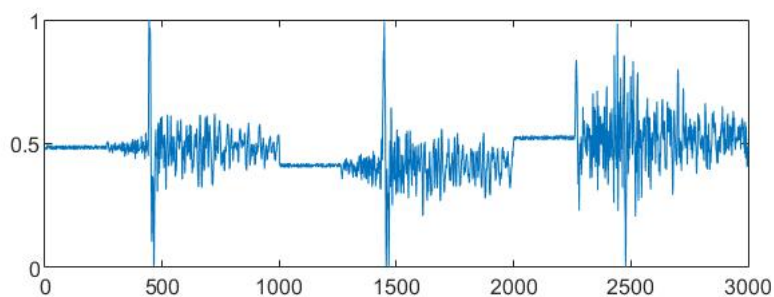


图 8: 增加最大值 2% 的噪音数据

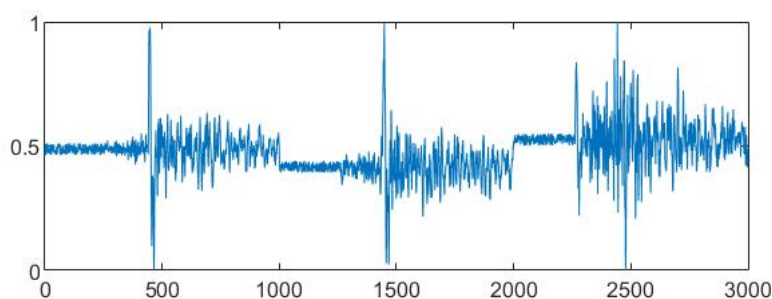


图 9: 增加最大值 5% 的噪音数据

```

data_waveSys2(:,i)=(data_waveSys2(:,i)-minvalue)/(maxvalue-minvalue);
14 end
16 % new_data_wave=[data_wave data_waveSys];
% new_data_cluster=[data_cluster data_cluster];
18
new_data_wave=[data_wave data_waveSys data_waveSys2];
20 new_data_cluster=[data_cluster data_cluster data_cluster];

```

可见，加入人工噪声的训练集其训练效果得到了十分显著的提高。最后可以通过工具箱生成对应的神经网络函数和训练脚本。

### 3 结果分析

我们利用前节中得到的神经网络函数对一段地震数据进行检测。我们检测 GS.OK029 台站 2014-02-16 00:00:00 到 2014-02-16 10:00:00 的数据，在地震目录中，这段时间一共有 4 个地震数据，利用神经网络可以检测出 9 个数据，具体情况如下表所示

使用的 Matlab 代码如下

```

for i=1:length(data_cluster)
2   data_waveSys(:,i)=data_wave(:,i)+0.02*max(data_wave(:,i))*rand
   (3000,1);
   minvalue=min(data_waveSys(:,i));
4   maxvalue=max(data_waveSys(:,i));
   data_waveSys(:,i)=(data_waveSys(:,i)-minvalue)/(maxvalue-minvalue);

```

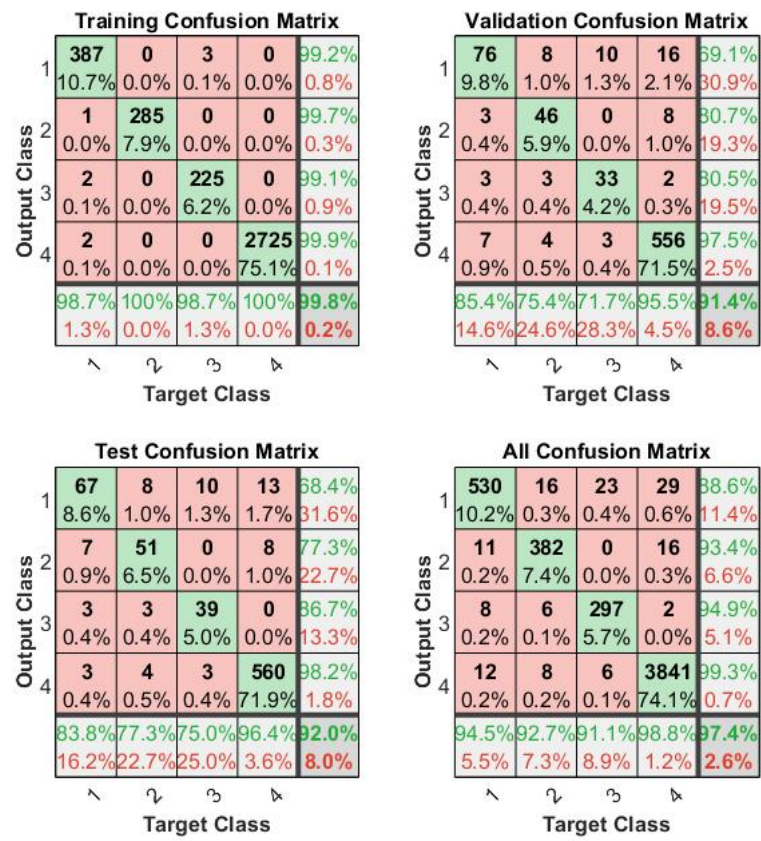


图 10: 训练数目为 5190 时的训练结果 (加入一套人工噪声数据集)

地震事件检测结果	
第一类事件个数	2
第二类事件个数	7
第三类事件个数	0
噪音事件个数	3591

表 1: 神经网络地震事件检测结果



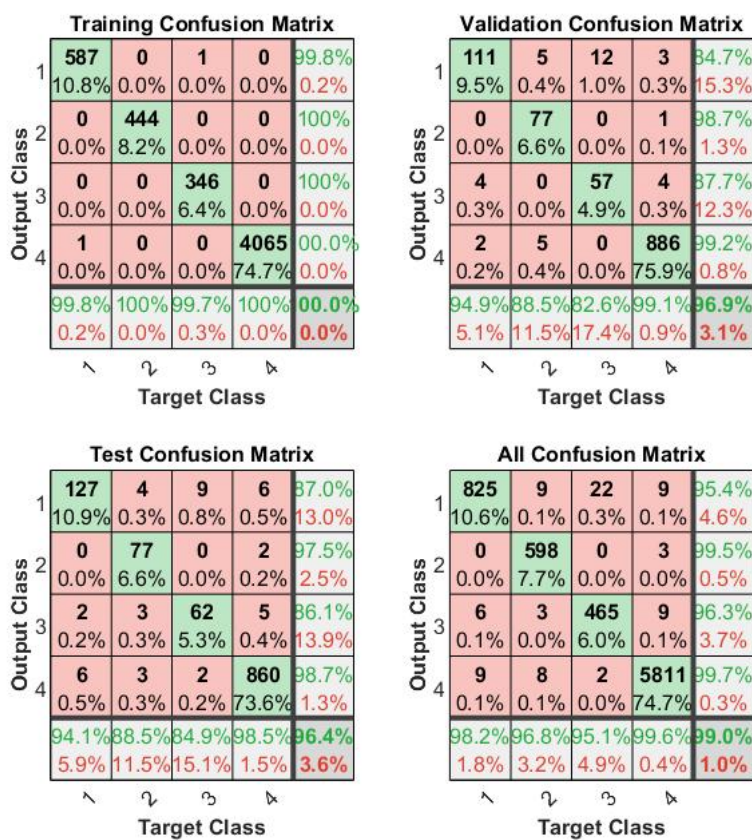


图 11: 训练数目为 7585 时的训练结果 (加入两套人工噪声数据集)

```

6   end
8
10  for i=1:length(data_cluster)
12      data_waveSys2(:,i)=data_wave(:,i)+0.05*max(data_wave(:,i))*rand
14      (3000,1);
16      minvalue=min(data_waveSys2(:,i));
18      maxvalue=max(data_waveSys2(:,i));
20      data_waveSys2(:,i)=(data_waveSys2(:,i)-minvalue)/(maxvalue-minvalue)
    ;
end

% new_data_wave=[data_wave data_waveSys];
% new_data_cluster=[data_cluster data_cluster];

new_data_wave=[data_wave data_waveSys data_waveSys2];
new_data_cluster=[data_cluster data_cluster data_cluster];

```

其中一些波形数据如下图所示

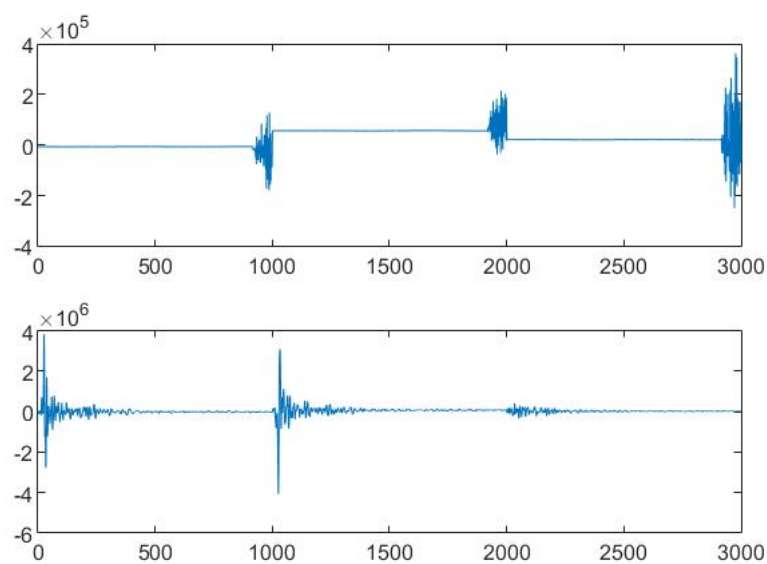


图 12: 部分检测事件波形数据

可见，实际上这是同一段波形（被分割成两段）。因此，此方法还需要

判断该波形究竟是一次地震还是因为时窗导致的两个“假事件”

## 4 总结与展望

### 4.1 结论

根据上述的成果，神经网络方法在一定程度上可以对地震事件进行检测。

### 4.2 研究过程发现的问题

可能由于神经算法工具箱的问题，最后识别的效率和结果和参考论文结果相去甚远，原因可能是数据集过小（文中使用了大约 60Gb 的数据集）

在整个研究过程中，我们也发现的研究中的许多问题。

1. 地震数据和噪声数据的选取是依赖于地震目录的。如果地震目录选取的不好，那么就会把一些可能蕴含着微小噪声的波形数据当作噪声数据送到神经网络进行训练了

2. 神经网络本身是一个“黑箱子”，参数的设置是没有任何物理意义的。即使训练的效果再好，我们也不能通过分析神经网络参数的方法得到有用的信息。

3. 本次训练的事件数目过少，只能通过对事件加入高斯白噪声的方法增加训练集，从而达到良好的训练效果。但是这样在一定程度上导致了过拟合现象的发生。

4. 在本次研究的最开始，我们打算研究自贡区域的地震检测，但是无法找到相关的数据，只能从 Iris 上查找其他有类似诱发地震的区域，也就是本文主要参考文献中的 Oklahoma 州地区。在国内，个人是很难得到波形数据的，这人为增加的研究成本。

5. 受限于笔记本电脑的性能，当神经网络层数和节点数过大时，就会有内存不足的现象发生。

6. 本次采用的时间窗长度为 10s，但是这个参数的选择是没有理由的。并且有时候截取波形的时候可能会截断了部分后续的面波信号。

### 4.3 展望与致谢

本次研究还缺少对传统地震事件检测方法的对比,如模板匹配法,FAST方法等。该神经网络方法也可以用于国内检测小型的诱发地震。

无论如何,随着运算效率的提高,类似于神经网络一类的机器学习方法必定会为人类的生产实践、科学研究提供巨大的帮助。对于地震学科也是如此。在这里特别感谢李远芳同学在研究过程中提供的帮助。李远芳帮助我解决了训练集不够的问题,并且为本次小型研究提供了许多帮助。