

语音识别期末项目——乐器识别

斜体字为注释，编辑时删去

项目组成员

学号	姓名
1952335	代玉琢
1953902	高杨帆
1851881	程凡

语音识别期末项目——乐器识别

- 一 项目要解决的问题
- 二、项目原理
 - 1.音色 (Timbre)
 - 2.快速傅里叶变换 (Fast Fourier Transform, FFT)
 - 3.频率移动 (Frequency Shifting)
 - 4.MFCC特征(Mel Frequency Cepstrum Coefficient, MFCC)
 - 5. 弹性反向传播(Resilient Backpropagation)
 - 6. 多层感知机(Multilayer Perceptron, MLP)
 - 7. 早期停止 (Early Stopping)
 - 8. MATLAB Neural Net Fitting
- 三 项目运行过程
 - 1. 数据预处理
 - 1.1 数据集的选择
 - 1.2 数据集的标签
 - 2.数据处理
 - 3. 神经网络训练
 - 4. 输入识别
- 四 项目表现评估
- 五 项目的优势及不足

一 项目要解决的问题

我们的项目聚焦于乐器识别。在一个交响乐团里，各种乐器的声音常常混合重叠在一起，如果不是专业的音乐人士或者未专门研究过，是很难准确地判断出各种音色是属于什么乐器的。

在我们的项目中，一共选择了八种乐器共五类音色，分别是民族乐器——五弦琴（班卓，banjo）；弦乐——大提琴（Cello）、小提琴（Violin）；哨片乐器——单簧管（Clarinet）、双簧管（Oboe）；铜管乐器——小号（Trumpet）；吉他（Guitar），钢琴（Piano）。

我们的项目聚焦于乐器识别，通过RBP（Resilient Back Propagation, 弹性反向传播）神经网络,多层感知机，和MATLAB神经网络Toolbox，以及伦敦爱乐乐团提供的数据集，实现了一个可以识别五弦琴（Banjo），大提琴（Cello），单簧管（Clarinet）,吉他（Guitar）,双簧管（Oboe），钢琴（Piano），小号（Trumpet）,小提琴（Violin）8种乐器的网络模型。

二、项目原理

由于各种乐器具有不同的音色，这种差异可以通过频域信号进行特征提取。所以我们需要通过快速傅里叶变换，将输入的时域信号转换为频域信号，进一步提取特征。

本项目在模型训练时利用RBP作为启发式学习函数，以早期停止的多层感知机（MLP）为模型进行神经网络的训练。

1.音色（Timbre）

音色（Timbre），是指不同声音的频率在波形方面与众不同的特性。不同的物体振动都有其特点。

声音是由发声的物体振动产生的。由于不同的振动总是可组合成为不同的声音，当物体振动时会发出基音，同时其各部分也有复合的振动，各部分振动产生的声音组合成泛音。所有的不同的泛音都比基音的波长短，且强度都相当弱，所以它盖不过比较强的基音，音乐家就可以调准乐器的音高。声音除的“基音”，加上许多不同“频率”（振动的物体1秒钟振动的次数）与泛音“交织”，就决定了不同的音色，使人听了以后能辨别出是不同的声音。每一种乐器、不同的人的声带，以及其它所有的能振动的物体都能够发出各有特色的不同的声音。

2.快速傅里叶变换（Fast Fourier Transform, FFT）

快速傅里叶变换 (fast Fourier transform), 即利用计算机计算离散傅里叶变换（DFT）的高效、快速计算方法的统称，简称FFT。快速傅里叶变换是1965年由J.W.库利和T.W.图基提出的。采用这种算法，能使计算机计算离散傅里叶变换所需要的乘法次数大为减少，特别是被变换的抽样点数N越多，FFT算法计算量的节省就越显著。

离散傅里叶变换（DFT）的公式：

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

其中，N为一帧的采样点个数。如，我们的采样频率为4000Hz，一帧取25ms，则 $N = 4000Hz \times 25ms$ 。

3.频率移动（Frequency Shifting）

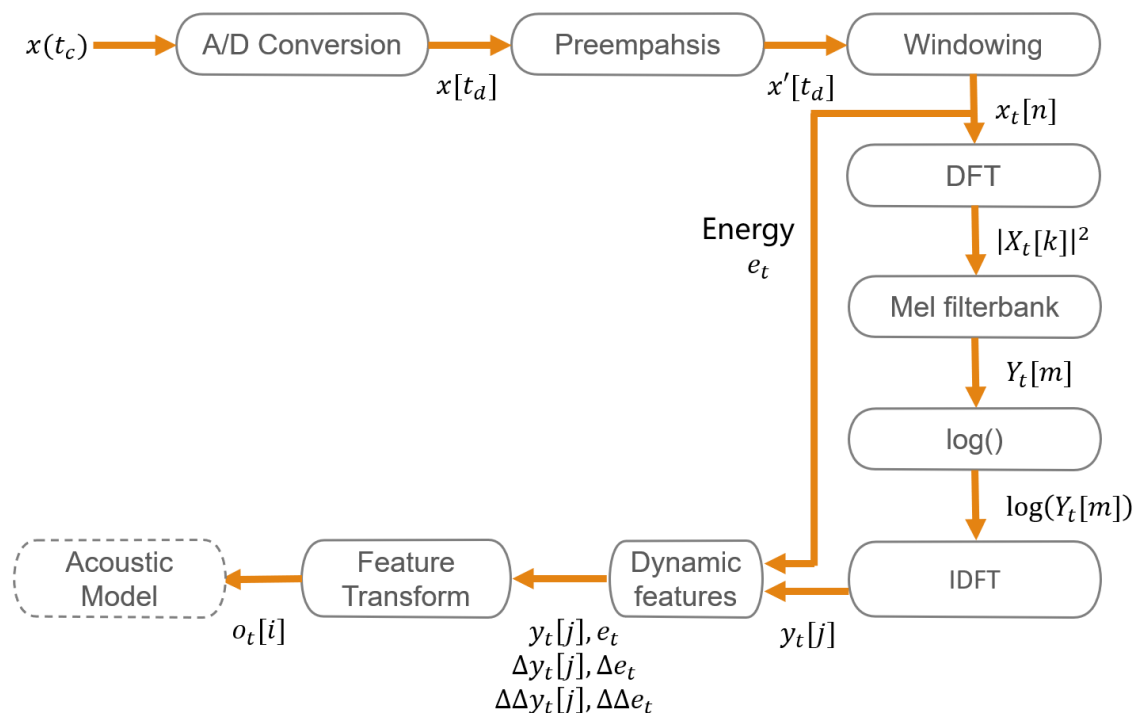
为了在模型训练时得到的特征是对乐器的特征进行分类而不是对音调的高低进行分类，所以我们需要消除音高对模型训练的影响，即将音调进行频率移动到同一个范围内。参考钢琴的音域标准，我们把不同基频的声音都移动到A4（440Hz）的基频下。

4.MFCC特征(Mel Frequency Cepstrum Coefficient, MFCC)

MFCC，即Mel频率倒谱系数（Mel Frequency Cepstrum Coefficient, MFCC）的缩写。Mel频率是基于人耳听觉特性提出来的，它与Hz频率成非线性对应关系。Mel频率倒谱系数(MFCC)则是利用它们之间的这种关系计算得到的Hz频谱特征。

MFCC已经广泛地应用在语音识别领域。由于Mel频率与Hz频率之间非线性的对应关系，使得MFCC随着频率的提高，其计算精度随之下降。因此，在应用中常常只使用低频MFCC，而丢弃中高频MFCC。

整个过程如下图所示。



5. 弹性反向传播(Resilient Backpropagation)

反向传播算法 (BP算法) :

反向传播算法, 简称BP算法, 适合于多层神经网络的一种学习算法, 它建立在梯度下降法的基础上。BP网络的输入输出关系实质上是一种映射关系: 一个n输入m输出的BP神经网络所完成的功能是从n维欧氏空间向m维欧氏空间中一有限域的连续映射, 这一映射具有高度非线性。它的信息处理能力来源于简单非线性函数的多次复合, 因此具有很强的函数复现能力。这是BP算法得以应用的基础。

反向传播算法主要由两个环节(激励传播、权重更新)反复循环迭代, 直到网络的对输入的响应达到预定的目标范围为止。

BP算法的学习过程由正向传播过程和反向传播过程组成。在正向传播过程中, 输入信息通过输入层经隐含层, 逐层处理并传向输出层。如果在输出层得不到期望的输出值, 则取输出与期望的误差的平方和作为目标函数, 转入反向传播, 逐层求出目标函数对各神经元权值的偏导数, 构成目标函数对权值向量的梯度, 作为修改权值的依据, 网络的学习在权值修改过程中完成。误差达到所期望值时, 网络学习结束。

弹性反向传播算法 (Resilient Backpropagation) :

正常使用的反向传播算法有两个缺点待解决, 其一为学习过程中学习率的选择较难, 一旦学习率选择不当会造成学习效果不好; 其二为反向传播算法的梯度弥散作用, 即距离输出层越远的神经元学习的速度越慢。

Martin Riedmiller因此提出了弹性反向传播算法。其中RBP中学习率 Δ 与传统的BP算法不同, 在执行过程中学习率会不断地进行更新。

RBP神经网络的权重更新函数为

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}$$

其中 $w_{ij}^{(t)}$ 表示权重, $\Delta w_{ij}^{(t)}$ 的值取决于模型偏导的符号, 如下式所示

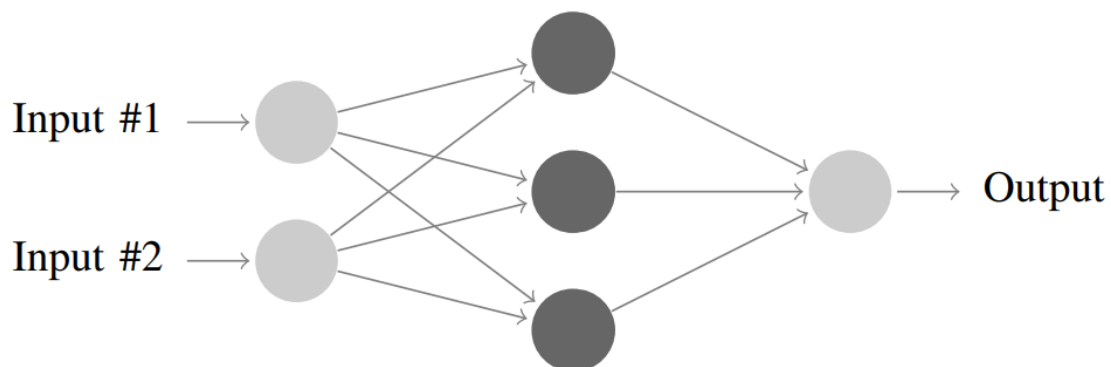
$$\Delta w_{ij}^{(t)} = \begin{cases} -\Delta_{ij}^{(t)} & , \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t)} > 0 \\ +\Delta_{ij}^{(t)} & , \text{ if } \frac{\partial E}{\partial w_{ij}}^{(t)} < 0 \\ 0 & , \text{ else} \end{cases}$$

其中 $\Delta_{ij}^{(t)}$ 表示学习率

这种学习算法可以使学习率更新的速率变快，从而加速模型训练的速率。

6. 多层感知机(Multilayer Perceptron, MLP)

多层感知机又叫ANN(人工神经网络)，除了输入输出层，还可以有多个隐藏层，一般所用的是三层感知机，示意图如下所示，假设输入层为向量 \mathbf{x} ，则隐藏层输出为 $f(w_i \mathbf{x} + b_i)$ ，其中 f 为激活函数， w_i, b_i 分别为权重和偏移量。



在本项目中，由于只是进行分类，所以 f 的选择为一个比较简单的符号函数

$$f_h(S) = \begin{cases} 1, & S > 0 \\ 0, & S \leq 0 \end{cases}$$

其中 S 就是上文的 $w_i \mathbf{x} + b_i$ ，具体为

$$S = \sum_{i=0}^n w_i x_i \quad \text{where} \quad \begin{cases} x_0 = -1 \\ w_0 = \theta \end{cases}$$

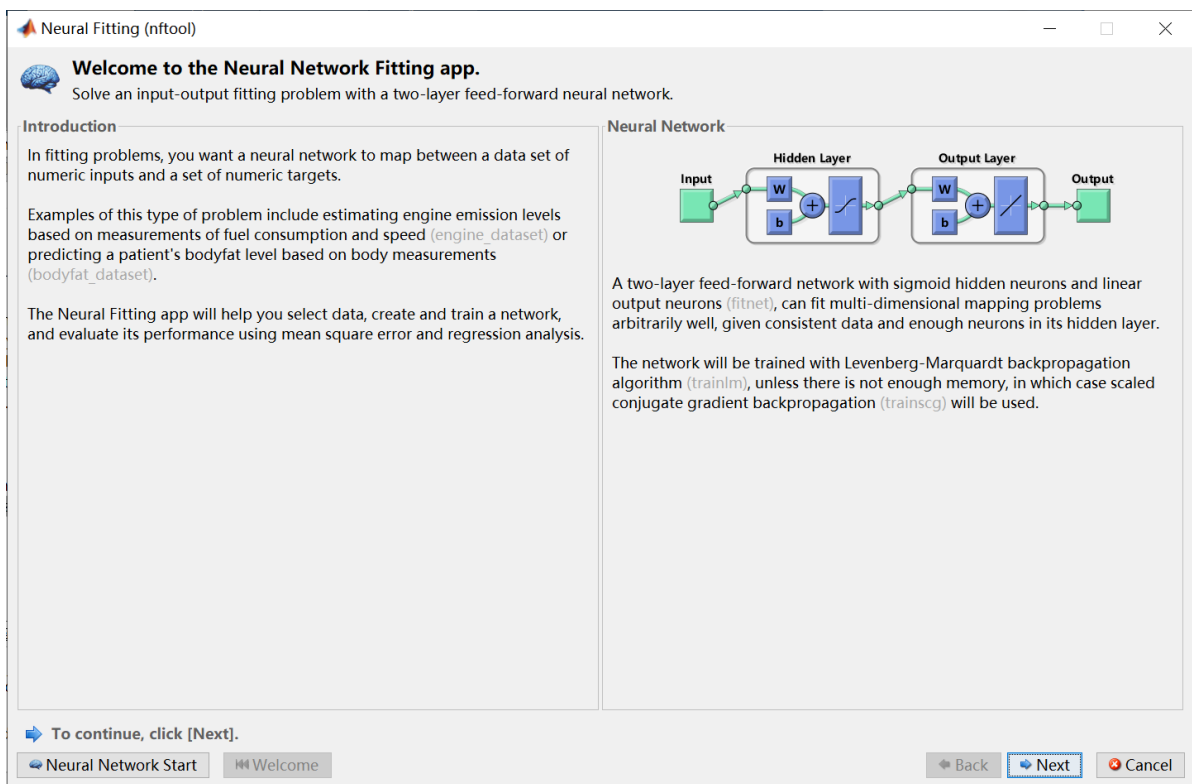
7. 早期停止 (Early Stopping)

早期停止技术是为了防止模型过拟合，它将数据集分为三个部分，除了原有的Training和Testing之外，还增加了一个Validation集（验证集）。用来限定每次训练错误的次数，在本项目中是150次，如果训练的过程中达到了这个错误次数，就立刻重新训练，最后从所有训练的模型中选出一个错误率最小的模型进行测试，作为一次训练模型的准确率。

8. MATLAB Neural Net Fitting

MATLAB是一款强大的数学软件，可用于数据分析、无线通信、深度学习、图像处理与计算机视觉、信号处理等领域。

在MATLAB中由许多神经网络工具箱，Neural Net Clustering用于聚类问题；Neural Net Fitting用于拟合问题；Neural Net Pattern Recognition用于模式识别问题等。



本项目主要使用MATLAB的Neural Net Fitting进行模型的训练。

Neural Net Fitting可创建并训练一个可视化的两层前馈网络来解决数据拟合问题。

使用该工具箱，输入数据，设置传递函数、训练函数、学习函数等建立一个前馈反向传播网络。再设置训练参数（如迭代次数、梯度、最大失败次数等）进行模型的训练。该工具箱将会定义和训练神经网络，并使用均方差和回归分析评估网络性能。使用可视化图分析结果，如回归拟合曲线。

三 项目运行过程

本项目对数据的处理主要分为以下几个步骤：

1. 数据预处理

1.1 数据集的选择

本项目使用的数据集由英国伦敦爱乐乐团音乐家录制的公开数据集（<https://philharmonia.co.uk/resources/sound-samples/>）及kaggle上的一个公开钢琴数据集组成。

各类样本数据集如下：

乐器	样本个数
banjo	23
cello	211
clarinet	226
guitar	37
oboe	193
piano	58
trumpet	169
violin	447
总计	1364

1.2 数据集的标签

为了统一生成数据集的特征，我们选择对数据集进行数据标记。具体格式为**instrument_tone_xxx**。其中instrument为该数据样本的乐器类型，tone为该数据样本的音调。

2. 数据处理

1. **FFT**,即快速傅里叶变换，将时域信号转换为频域信号。
2. **频谱切割**，保留前1000Hz的数据。
3. **频率移动**，由于本项目聚焦于乐器识别，即聚焦于分类，音调的高低会对音色的识别带来影响，故我们对音频进行了移动，使其可以用A4(440Hz)的音符表示。
4. **特征提取**，特征提取之前，我们将频谱分成了50块，这是为了防止过拟合，后面我们通过求每一个块中的平均频率值来提取特征。
5. **归一化**，这是为了平均化特征，使得每一个采样对特征的贡献相等，归一化函数如下：

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

经过这些步骤之后，我们将样本的特征保存为一个.mat文件

3. 神经网络训练

我们利用上一步处理过的数据来进行神经网络的训练，其中，训练函数由MATLAB的神经网络Toolbox提供，我们只需调整参数，调用即可。

在网络创建时使用的函数是NEWFF(P,T,S,TF,BTF,BLF,PF,IPF,OPF,DDF)。

其中：

P：输入参数矩阵(RxQ1)，每一列都是一个样本。在这里即为输入的样本的特征向量。

T：目标参数矩阵。在这里即我们的八维向量组成的矩阵。

S：N-1各隐含层的数目。

TF：相关层的传递函数，在这里我们设置的隐含层为tansig函数，即正切S型传递函数，输出层为logsig函数，即对数S型传递函数。

BTF：BP神经网络的学习训练函数。由于本项目使用的是弹性反向传播算法，所以我们设置的是trainrp。

BLF: 权重学习函数。

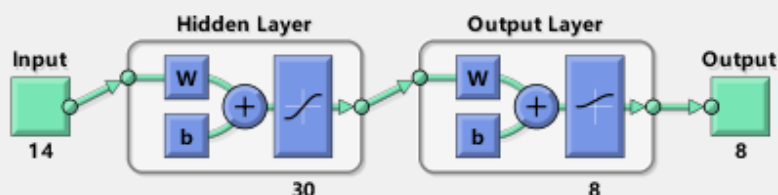
PF: 性能函数, 设置为mse, 即均方差。

IPF、OPF、DDF为默认值。

```
net = newff(input_data, mapped_label_data, [30], {'tansig' 'logsig'},  
'trainrp', '', 'mse', {}, {}, 'dividerand');  
  
% INIT Initialize a time series object with new time and data values  
% 用新的时间和数据值初始化一个时间序列对象  
net = init(net);  
  
% Custom parameters  
% 设置训练参数  
net.trainParam.epochs = 500; % 训练最大次数  
net.trainParam.lr = 0.1;  
net.trainParam.min_grad = 0;  
net.trainParam.max_fail = 150; % validation 最大容错量  
  
% Train network  
% 训练网络  
% input_data-网络实际输入, mapped_label_data-网络应有输出  
  
[trained_net, stats] = train(net, input_data, mapped_label_data);  
% trained_net:训练得到的网络
```

一次网络训练过程截图如下:

Neural Network



Algorithms

Data Division: Random (dividerand)

Training: RProp (trainrp)

Performance: Mean Squared Error (mse)

Calculations: MEX

Progress

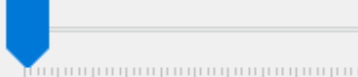
Epoch:	0	464 iterations	500
Time:		0:00:00	
Performance:	0.387	0.0426	0.00
Gradient:	0.208	0.00137	0.00
Validation Checks:	0	150	150

Plots

Performance (plotperform)

Training State (plottrainstate)

Regression (plotregression)

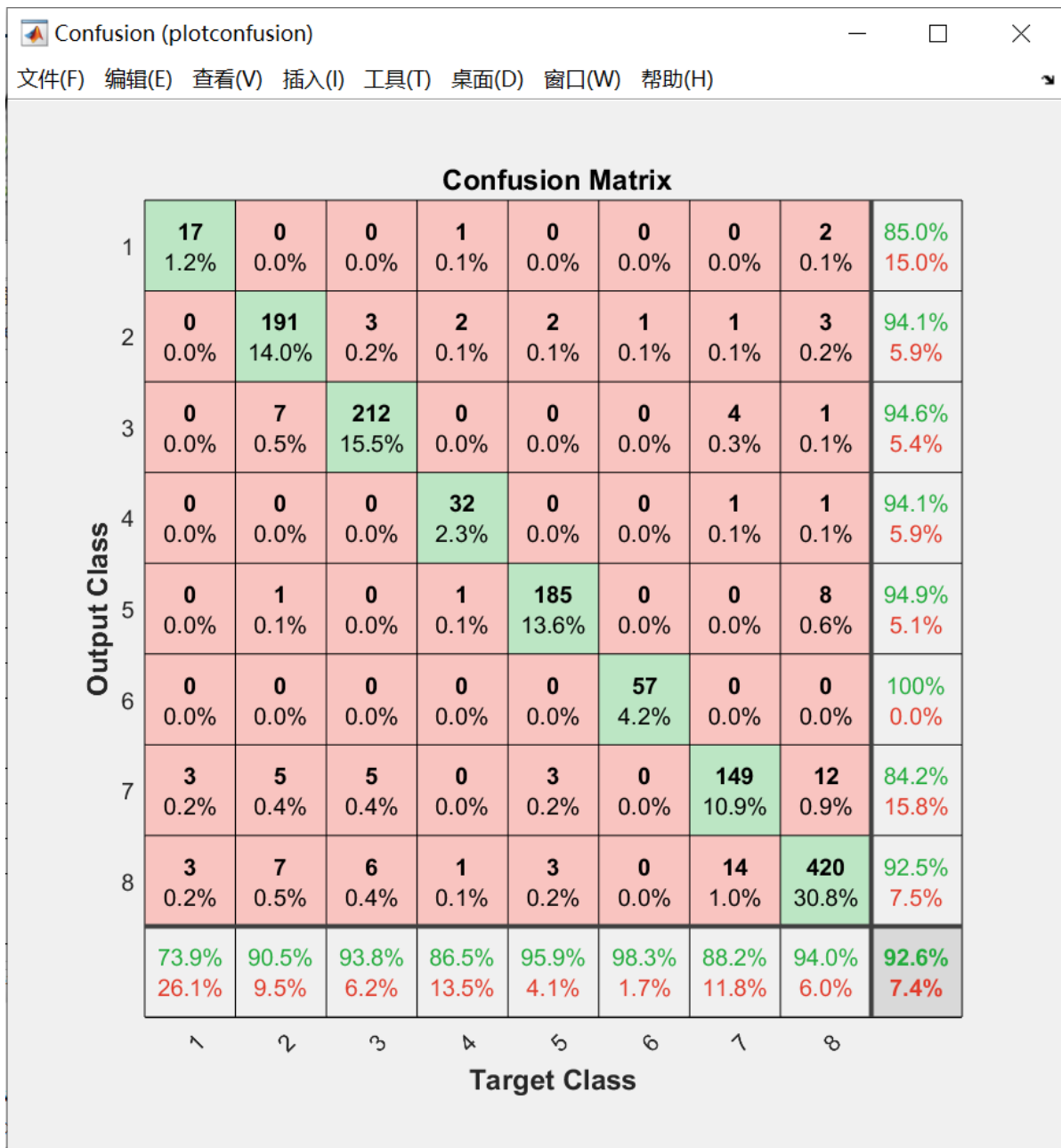
Plot Interval:  1 epochs

✓ **Validation stop.**

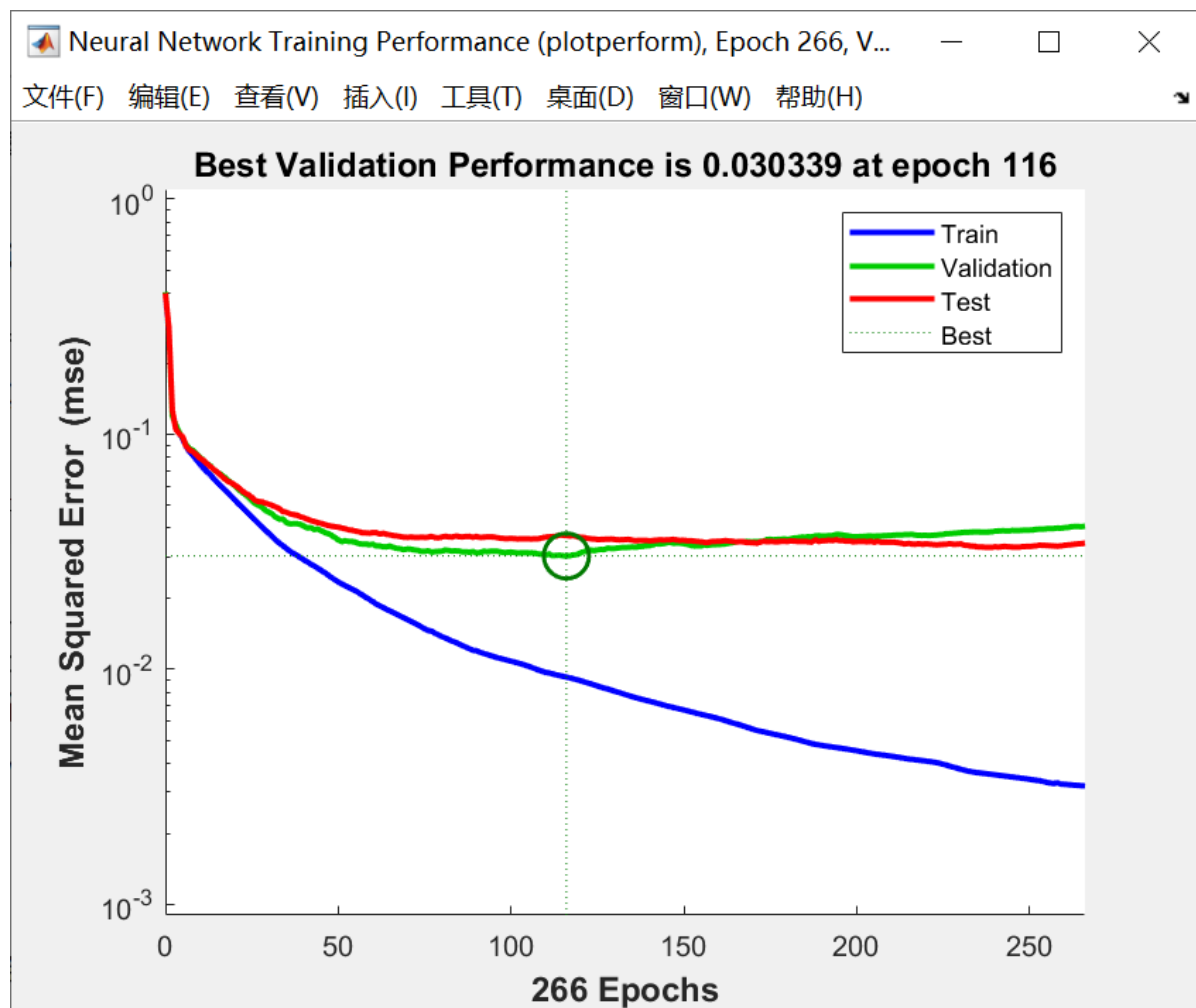
● Stop Training

● Cancel

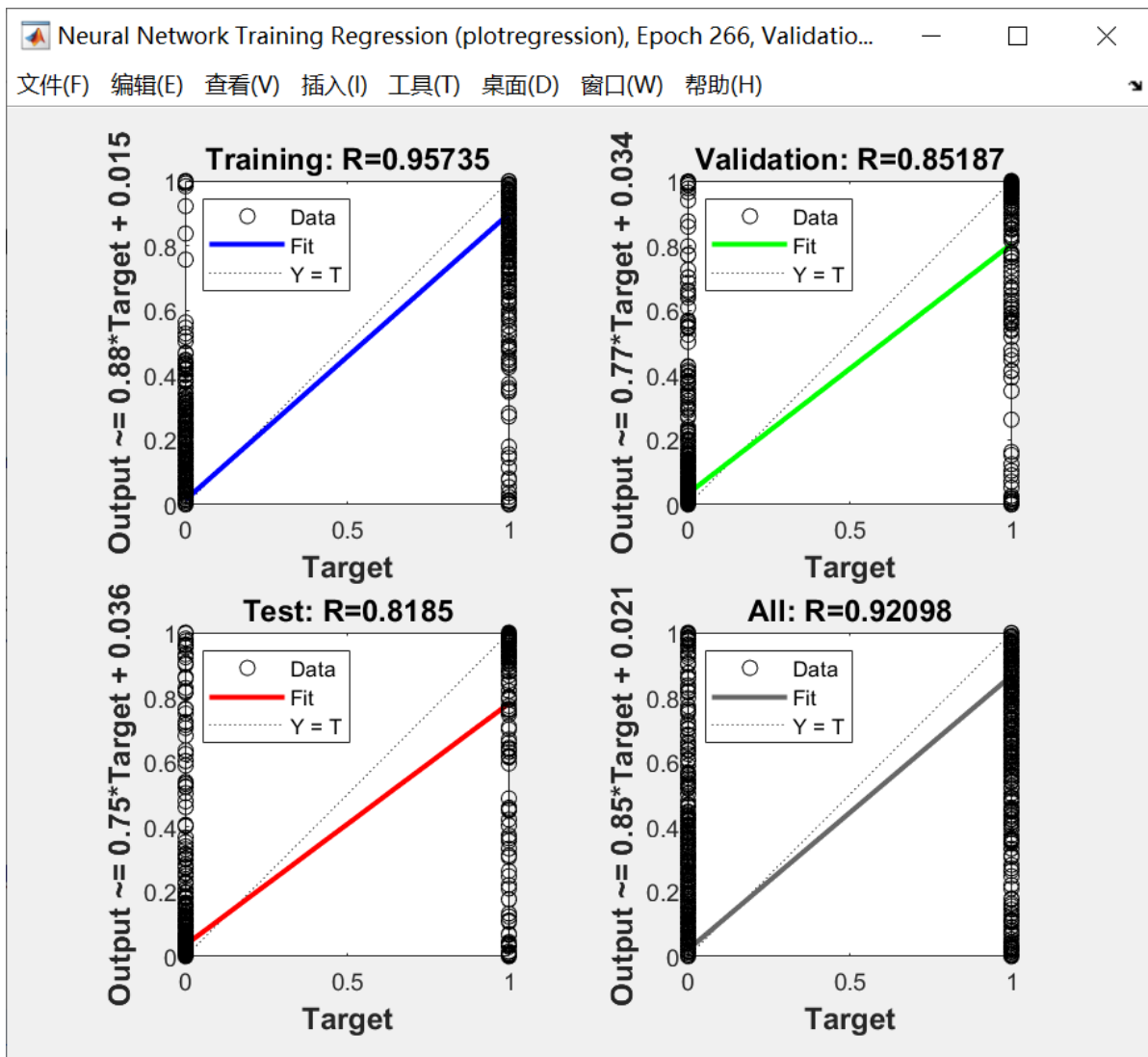
一次网络训练混淆矩阵如下:



Performance:



Regression:



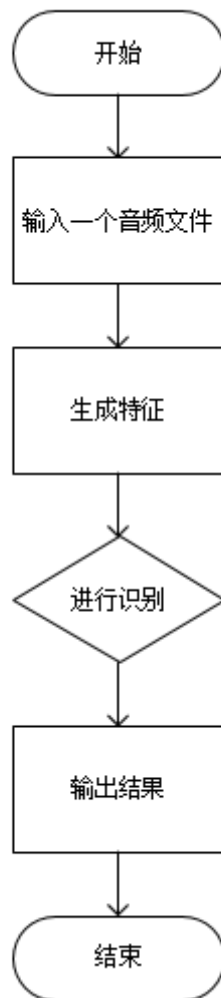
我们使上述训练过程运行100次，从中选取准确率最高的模型保存下来

```
if accu>accuracy
    save('model_mfcc','trained_net','accu');
    accuracy = accu;
    fprintf('saved model accuracy:%f\n',accu);
end
```

4. 输入识别

输入识别分为以下步骤：

流程图：



1. **进行数据处理中的各步骤**，将音频文件转化为网络能够识别的信号
2. **调用训练好的网络进行识别**，代码如下

```
predicted = trained_net(input_data)
```

3. **对识别结果进行处理**，上述代码的输出是一个1*8的向量，分别对应上述8种乐器的评分，评分最高者对应的索引即为预测结果

```
[max_t,index_t]=max(predicted);
```

4. **输出最后的识别结果**，代码如下

```
instrument=[  
"banjo","cello","clarinet","guitar","oboe","piano","trumpet","violin"];  
fprintf('识别结果: %s \n',instrument(index_t));
```

四 项目表现评估

按照原有的特征提取方式，模型的准确率最高可达到95.45%，如下图accu属性所示：

	accu	95.4545
	trained_net	object

同时我们的项目还支持用户输入文件名，输出识别结果，示例如下：

请输入文件名(输入-1停止)：

piano_G4_miku.wav

识别结果：piano

预期结果：piano

其中，文件命名规范已在 `recognition.m` 中给出。

五 项目的优势及不足

优势：

1. RBP神经网络比较简单，易于实现，且训练速度快。
2. 模型的准确率高，比其他的算法训练出的模型效果好。

在创建神经网络时，我们选择了多种学习训练函数进行测试，包括trainoss，即OSS反向传播训练算法，准确率只有83.43%；traingdx，自适应调节学习率并附加动量因子的梯度下降反向传播算法训练函数，准确率只有72.8%；trainlm，LM反向传播算法训练函数，训练时间最长但是准确率也只有85.4%。

3. 提供了输入接口，用户可以输入wav格式的音频文件，进行乐器识别，同时还提供了降噪接口，如果用户输入的音频噪声较大，则可以选择降噪输入以提高识别的准确度

不足：

1. 目前的网络对小提琴的识别准确率较低，如下所示

请输入文件名(输入-1停止)：

violin_E4_SS.wav

识别结果：clarinet

预期结果：violin

针对这个不足，我们采用**MFCC特征**代替原有特征进行了处理和训练，虽然模型的整体准确率有较大的下降，如下accu属性所示

	accu	77.4194
	trained_net	object

但是对于小提琴的识别精度提高了，如下所示

```
violin_E4_SS.wav
识别结果: violin
预期结果: violin
请输入文件名(输入-1停止):
piano_E4_SS.wav
识别结果: piano
预期结果: piano
```

同时对于其他乐器的识别没有显著的下降。

2. 我们的降噪接口在我们的机器上测试时，由于设备录音时收音效果较差，故降噪之后识别准确率较低。

参考文献

Babak Toghiani-Rizi, Marcus Windmark. *Musical Instrument Recognition Using Their Distinctive Characteristics in Artificial Neural Networks*[EB/OL]. 2017[2021.12.24]. <https://arxiv.org/abs/1705.04971>.