

第一周报告 - 于建国 (YJango)

1. 特征提取
2. 预处理
3. 模型选择
 - 3.1. 选择策略
 - 3.2. 浅层模型结果
 - 3.2.1. 单特征集
 - 3.2.1. 多特征集
4. 结论
5. 附：导师评语
 - 5.1. 导师1
 - 5.2. 导师2

第二周报告 - 于建国 (YJango)

1. 特征提取
2. 特征筛选
 - 2.1. 全局特征筛选
 - 2.1.1. 全连接网络结构
 - 2.1.2. 全连接网络 7-fold CV
 - 2.1.3. 全局特征选择
 - 2.2. 时序特征筛选
 - 2.2.1. 一维卷积网络结构
 - 2.2.2. 一维卷积网络 7-fold CV
 - 2.2.3. 双向循环网络结构
 - 2.2.4. 双向循环网络 7-fold CV
 - 2.2.5. 时序特征选择
3. 特征融合
 - 3.1. 网络结构
 - 3.2. 网络结构 7-fold CV
4. 代码说明
 - 4.1. 特征提取
 - 4.2. 神经网络
5. 线上测试结果
6. 结论
7. 附：导师评语
 - 7.1. 导师1
 - 7.2. 导师2

第一周报告 - 于建国 (YJango)

1. 特征提取

特征	维度
IS09_emotion.conf	384
IS10_paraling.conf	1582
IS13_ComParE.conf	6373
ComParE_2016.conf	6373

extractor.py

- 作用：批量提取特征
- 格式：.wav → .txt

data_maker.py

- 作用：与标签对齐后转成 `pandas.DataFrame`
- 格式：.txt → .csv

2. 预处理

标准化：

- x (特征): $x = (x - x.mean) / x.std$
- y (标签): $y = (y - y.mean) / y.std$

3. 模型选择

3.1. 选择策略

对 4 组特征集分别进行 7 折交叉验证，找出最好的特征和模型

3.2. 浅层模型结果

shallow.py

标准化后的结果 (没有反标准化)

3.2.1. 单特征集

特征	算法	mse	r^2
IS09	Support Vector Regression	0.5666	0.4324
IS09	Gradient Boosting Decision Tree	0.5402	0.4586
IS09	AdaBoost	0.6396	0.3678
IS09	Ridge Regression	0.6698	0.3272
IS09	Bayesian Ridge Regression	0.6082	0.3901
IS10	Support Vector Regression	0.4821	0.5164
IS10	Gradient Boosting Decision Tree	0.4645	0.5340
IS10	AdaBoost	0.5586	0.4452
IS10	Ridge Regression	2.550	1.562
IS10	Bayesian Ridge Regression	0.5484	0.4491
IS13	Support Vector Regression	0.5060	0.4923
IS13	Gradient Boosting Decision Tree	0.4591	0.5392
IS13	AdaBoost	0.5422	0.4498
IS13	Ridge Regression	0.8717	0.1241
IS13	Bayesian Ridge Regression	0.8723	0.1235
IS16	Support Vector Regression	0.5060	0.4937
IS16	Gradient Boosting Decision Tree	0.4693	0.5292
IS16	AdaBoost	0.5390	0.4592
IS16	Ridge Regression	0.8786	0.1175
IS16	Bayesian Ridge Regression	0.8791	0.1170

3.2.1. 多特征集

特征	维度	mse	r^2
IS10,13	Gradient Boosting Decision Tree	0.4484	0.5484
IS09,10,13	Gradient Boosting Decision Tree	0.4446	0.5517
IS09,10,13,16	Gradient Boosting Decision Tree	0.4428	0.5534

4. 结论

- 四个特征集全部使用效果最好
- 浅层模型 Gradient Boosting Decision Tree 效果最好
 - `num_leaves =15`
 - `learning_rate=0.02`
 - `n_estimators=400`
- 预测代码: `predict.py`
- 预测值为 GBDT 两次 (`n_estimators=400` 和 `n_estimators=800`) 的平均数

5. 附：导师评语

5.1. 导师1

- 项目完成情况很好，并对其过程进行了详细的说明。语音情感特征提取部分完成了 `IS09_emotion.conf`、`IS10_paraling.conf`、`IS13_ComParE.conf` `ComParE_2016.conf` 四个特征提取工作。而且完成了批量的特征提取。
- 数据预处理部分，完成了特征和标签的预处理。
- 对 4 组特征集分别进行 7 折交叉验证，找出最好的特征和模型。尝试了机器学习所有的几乎所有的模型，并且用多个指标进行验证，背后具有很多的工作量。
- 于建国同学在短短一周之内做出了很多模型和特征的尝试工作，其工程量庞大且很好完成任务。值得每位同学学习。

5.2. 导师2

- 该学员的学习报告对特征提取，预处理，模型选择等模块进行详细说明。文中多处使用表格，条理非常清晰。
- 用表格列举并对比了不同的特征，并说明了特征提取的过程，简述了特征提取后的格式转换过程。
- 预处理部分用到了均值方差归一化的小策略。
- 模型部分，对不同的特征集进行了详细的实验和说明。分别在单特征集和多特征集上进行多次实验，并详细列举不同特征集的表现。
- 最后根据不同特征集和不同模型的表现选出最好的性能，效果比较理想。

第二周报告 - 于建国 (YJango)

1. 特征提取

提取 OpenSMILE 全局特征 和 时序特征 后做成 `tfrecord` 便于特征的随意组合和 `tensorflow` 的训练。

全局特征	维度	时序特征 (l1d)	维度
IS09_emotion.conf	384	MFCC13*3	$timestep \times 39$
IS10_paraling.conf	1582	IS10_paraling -D	$timestep \times 76$
IS13_ComParE.conf	6373	IS13_ComParE -D	$timestep \times 130$
ComParE_2016.conf	6373	FilterBank(40+total energy)*3	$timestep \times 123$

2. 特征筛选

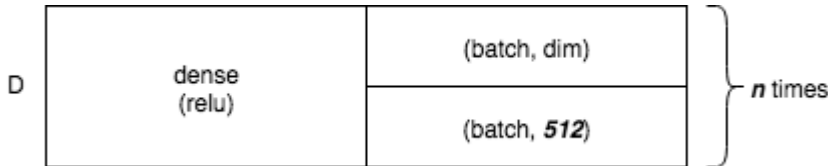
选择出最好的 全局特征 和 时序特征 用于做特征融合。

2.1. 全局特征筛选

使用全连接网络进行训练，取最好的网络结构的最后隐藏层作为全局特征。

2.1.1. 全连接网络结构

省略输入层和输出层，D 表示 dense，*n* 表示层数



全连接网络结构

layer	node	learning rate	batch size	epoch	update
5	512	5e-4	96	50	Adam

2.1.2. 全连接网络 7-fold CV

用全连接网络 (FC) 的结果和 Gradient Boosting Decision Tree 基线结果进行对比。

特征	维度	GBDT rmse	GBDT pcc	FC rmse	FC pcc
IS09	384	0.7349	0.6772	0.7539	0.6498
IS10	1582	0.6815	0.7307	0.6869	0.7214
IS13	6373	0.6775	0.7343	0.6741	0.7328

IS16	6373	0.6850	0.7274	0.6797	0.7294
IS09,13	384+6373	0.6729	0.7342	0.6748	0.7335
IS10,13	1582+6373	0.6763	0.7353	0.6616	0.7456
IS09,16	384+6373	0.6749	0.7326	0.6787	0.7306
IS10,16	1582+6373	0.6618	0.7453	0.6620	0.7459
IS09,10,13	384+1582+6373	0.6739	0.7339	0.6685	0.7386
IS09,10,13,16	384+1582+6373+6373	0.6639	0.7429	0.6697	0.7382

2.1.3. 全局特征选择

使用 IS10,13 和 IS10,16 特征集的全连接网络效果较好。

IS09,10,13,16 在全连接网络中并不容易训练。

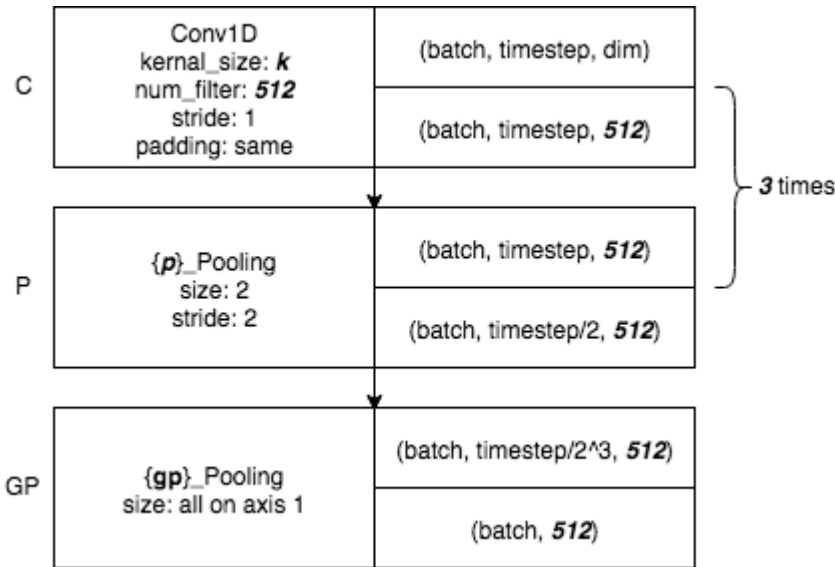
由于很多论文偏向于 IS13，所以选择使用 IS10,13 的全连接网络的最后隐藏层的输出 (512 维) 做为全局特征。

2.2. 时序特征筛选

尝试 一维卷积网络结构 和 双向循环神经网络 来建模。取最好的网络结构的最后隐藏层作为时序特征。

2.2.1. 一维卷积网络结构

卷积网络速度快，特征集的筛选主要由它负责。 k 是 kernel size, p 是 pooling type, gp 是在时间维上的平均池化。



一维卷积网络结构

pool size	learning rate	batch size	epoch
-----------	---------------	------------	-------

2	5e-5	32	100
---	------	----	-----

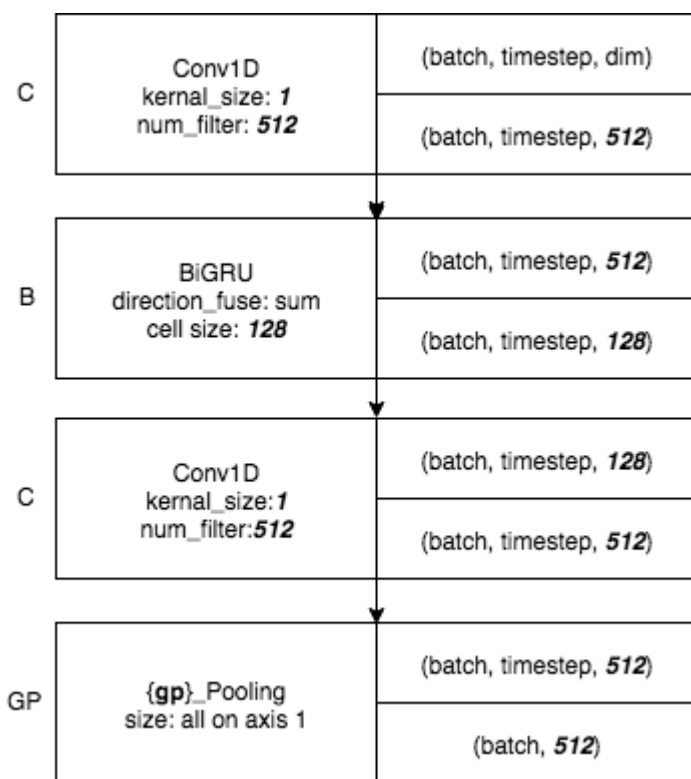
2.2.2. 一维卷积网络 7-fold CV

时序特征	维度	pool type	kernel size	<i>rmse</i>	<i>pcc</i>
mfcc	39	max	2	0.7503	0.6524
fb	123	max	2	0.7321	0.6716
mfcc + fb	39+123	max	2	0.7001	0.7059
mfcc	39	avg	2	0.7698	0.6272
fb	123	avg	2	0.7462	0.6545
mfcc + fb	39+123	avg	2	0.6969	0.7090
mfcc	39	max	3	0.7471	0.6558
fb	123	max	3	0.7297	0.6764
mfcc + fb	39+123	max	3	0.7117	0.6725
fb +IS10+IS13	39+76+130	max	2	0.6708	0.7348
fb +IS10+IS13	39+76+130	max	3	0.6736	0.7325

其余结果略

2.2.3. 双向循环网络结构

与循环神经网络进行对比，时间缘故，并未调节卷积层的 kernel size，只用了 1x1 conv。



双向循环网络结构

learning rate	batch size	epoch
1e-4	64	50

2.2.4. 双向循环网络 7-fold CV

时序特征	维度	<i>rmse</i>	<i>pcc</i>
mfcc 39	39	0.7110	0.6949
fb, IS10, 13	123+76+130	0.6888	0.7175

其余结果略

2.2.5. 时序特征选择

一维卷积网络稍优于，且快于循环网络结果。

特征上，单独的 mfcc 或 fb 都不如 IS10, 13 的特征集。

所以选择使用 fb, IS10,13 的循环 或 卷积网络的最后隐藏层的输出 (512 维) 做为时序特征。

3. 特征融合

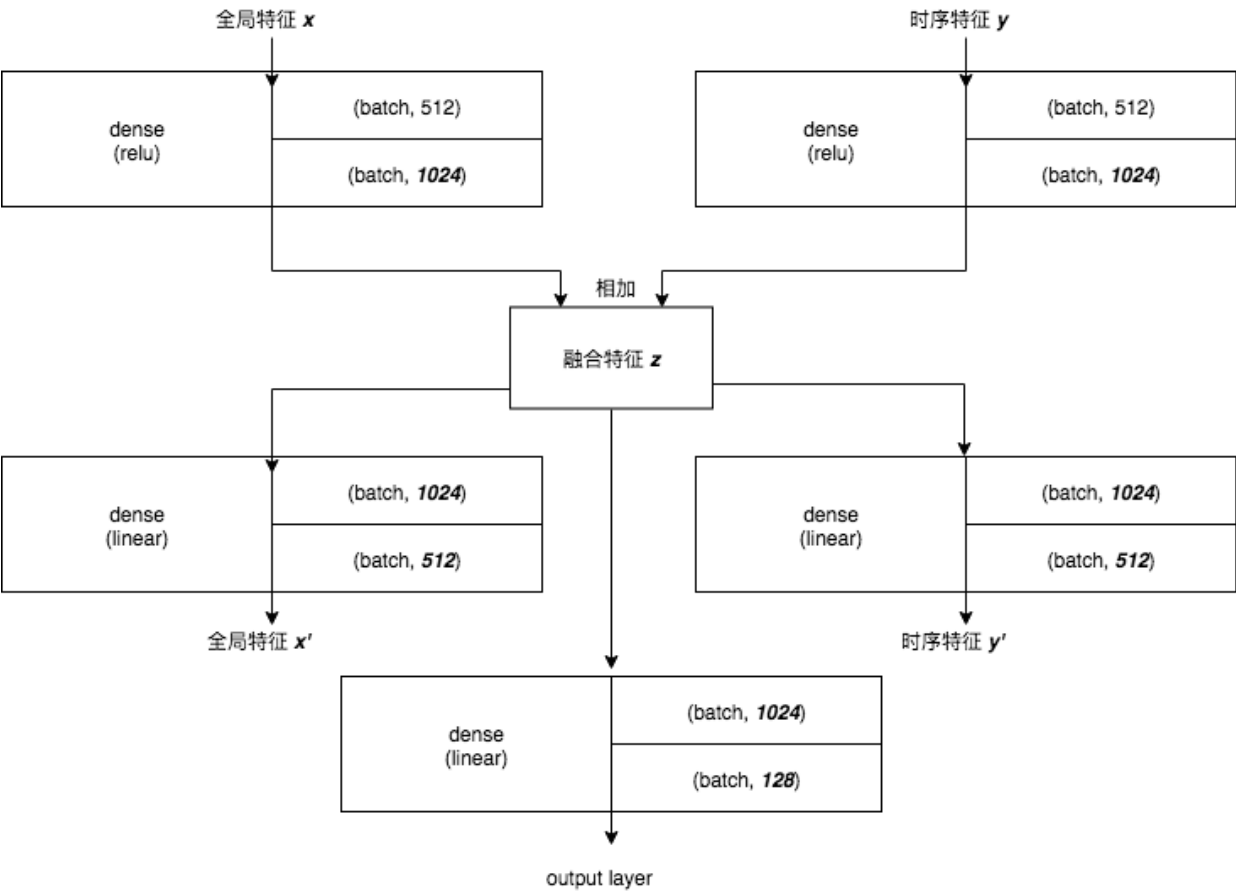
因为全局特征和时序特征只是形式不同，但有极强的相关性和重复信息。

所以使用 [correlational neural networks](#) 进行特征融合。

优点是仅使用一个输入 x (全局特征) 或 y (时序特征) 进行预测而不会过分破坏表现。

3.1. 网络结构

全局特征 和 时序特征的网络的权重是预训练后固定的。
训练完 correlational neural networks 后，固定网络权重，取 z 作为最后的特征。
最后把 z 作为输入，只训练 dense (relu) 和 output layer (共同形成MLP) 预测 P 值。



3.2. 网络结构 7-fold CV

模型	特征	维度	<i>rmse</i>	<i>pcc</i>
融合特征 using MLP	全局特征 (512) + 循环网络时序特征 (512)	512	0.6486	0.7569
融合特征 using MLP	全局特征 (512) + 卷积网络时序特征 (512)	512	0.6375	0.7655
mean (GBDT + 融合特征 using MLP)	IS10, IS16 (384+6373) + 全局特征 (512) + 卷积网络时序特征 (512)	512	0.6335	0.7692

4. 代码说明

4.1. 特征提取

代码文件	作用	格式变换
<code>extractor.py</code>	批量提取全局、时序特征	<code>.wav</code> → <code>.txt</code> or <code>.mfc</code>
<code>data_maker.py</code>	与标签对齐后转成 <code>pandas.DataFrame</code>	<code>.txt</code> → <code>.csv</code>
<code>tfrecord_maker.py</code>	将数据转成 <code>tfrecord</code> 文件	<code>.csv</code> or <code>.mfc</code> or <code>.txt</code> → <code>.tfrecord</code>
<code>data.py</code>	帮助读写 <code>tfrecord</code> 文件	null

4.2. 神经网络

代码文件	作用
<code>DNN.py</code>	全连接网络实验
<code>RNN.py</code>	循环网络实验
<code>CNN.py</code>	卷积网络实验
<code>Cr2NN.py</code>	相关网络实验
<code>layers.py</code>	搭建层的帮助文件
<code>cv_ids.npy</code>	交叉验证的文件 ID

5. 线上测试结果

在额外测试集上表现。

模型	<i>pcc</i>
GBDT	0.7342
全连接网络	0.7118
卷积网络	0.7280
融合特征 using MLP	0.7499
融合特征 using GBDT	0.7472
<code>mean</code> (GBDT + 融合特征 using MLP)	0.7544
<code>mean</code> (GBDT + 融合特征 using GBDT)	0.7546

6. 结论

此任务中：

- 特征
 - 全局特征 IS10,13 效果较好。
 - 时序特征 fb 优于 mfcc。
 - 时序特征 fb, IS10,13 效果较好。
 - 融合特征带来的提升比较显著，即使用来融合的 卷积模型特征 和 全连接模型特征本身效果很一般。
- 模型
 - 一维卷积网络速度、参数量、效果都略优于循环网络。
 - 融合特征不论是用神经网络还是 GBDT 都可以取得较好的结果。
 - 平均第一周模型的结果和第二周模型的结果取得了最好的结果。

7. 附：导师评语

7.1. 导师1

- 在特征提取上继续了上周的工作，并在此基础上筛选了全局特征和时序特征，并给出了筛选的过程。
- 在网络结构上进行了7折交叉验证，并与 GBDT 进行对比。
- 时序特征选择上使用一维卷积，双向LSTM。最后给出了融合特征的网络结构图。
- 在项目工程文档里面给出每个文件的说明，最终给出了结论。
- 并思考了深度学习网路结构在该数据集上的一些优缺点。

7.2. 导师2

- 该学员在报告伊始给出目录，报告结构一目了然。
- 中间过程多次使用表格对不同的项目进行对比，结构非常清晰，条理性非常强，逻辑严谨，思路清晰，描述详细。
- 文中给出了网络结构图，非常清晰。
- 对不同网络结构和特征描述非常详细，分析透彻。
- 并在最后给出自己的感悟和结论。