

# 基于 AI 的心电信号辅助诊断

精 83 2018010650 高易远

2018010637 潘益萱

## 目录

1 数据预处理 .....	2
2 构建 Dataset .....	4
3 模型 .....	5
3.1 基于卷积的模型 .....	5
3.1.1 ResNet .....	5
3.1.2 ResNeXt .....	5
3.1.3 SEResNet .....	6
3.1.4 ECGNet .....	6
3.2 基于序列的模型 .....	9
3.2.1 LSTM_V1 .....	9
3.2.2 基于 8 分支的 LSTM_V2 .....	10
3.2.3 基于 8 分支的 GRU .....	10
3.3 基于卷积和序列的模型 .....	13
4 训练, 验证和测试 .....	14
5 结果 .....	15
5.1 基于卷积的模型 .....	15
5.2 基于序列的模型 .....	16
5.3 基于卷积和序列的模型 .....	18
5.4 结果分析 .....	18
6 改进 .....	20

## 1 数据预处理

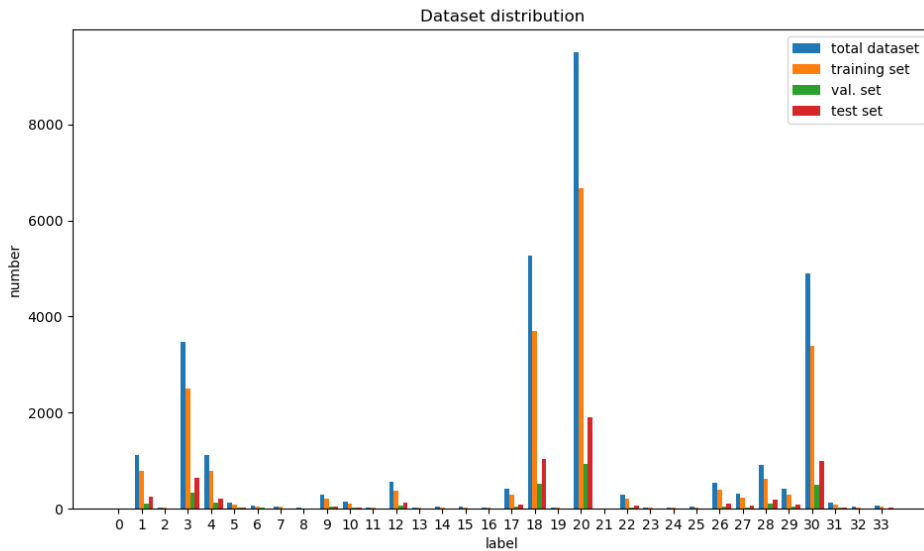
这部分代码见 `data_process.py`，把可视化的部分注释掉了。

踩过的一个坑是，如果数据装填在 `set` 中，一般是乱序的（每次都有不同），转成 `list` 之后需要 `sort()` 之后再处理。

处理结果如下，一共有 34 类异常，各类异常的数量分布十分不均匀，最多的是 20: '窦性心律'，有 9501 个数据；最少的是 0: 'QRS 低电压'，只有 3 个数据。

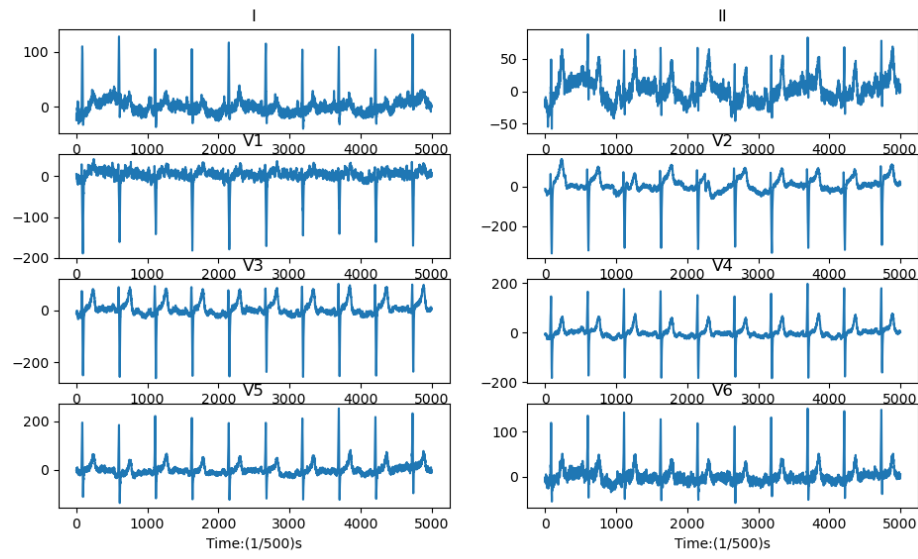
通过设置合适的随机数种子，划分后测试集中每种异常事件至少包含一个样本。

```
len of total dataset: 20036
len of train dataset: 14024
len of val. dataset: 2004
len of test dataset: 4008
label:
{0: 'QRS低电压', 1: '电轴右偏', 2: '起搏心律', 3: 'T波改变', 4: '电轴左偏', 5: '心房颤动', 6: '非特异性ST段异常', 7: '下壁异常Q波',
8: '前间壁R波递增不良', 9: 'ST段改变', 10: '一度房室传导阻滞', 11: '左束支传导阻滞', 12: '右束支传导阻滞', 13: '完全性左束支传导阻
滞', 14: '左前分支传导阻滞', 15: '右心房扩大', 16: '短PR间期', 17: '左心室高电压', 18: '窦性心动过缓', 19: '早期复极化', 20: '窦性心
律', 21: '融合波', 22: 'ST-T改变', 23: '非特异性ST段与T波异常', 24: '快心室率', 25: '非特异性T波异常', 26: '室性早搏', 27: '房性早搏',
28: '窦性心律不齐', 29: '完全性右束支传导阻滞', 30: '窦性心动过速', 31: '不完全性右束支传导阻滞', 32: '顺时针转位', 33: '逆时针转
位'}
label distribution of the total dataset:
[ 3 1124 16 3479 1124 120 64 52 16 286 142 25 551 25
 35 32 23 414 5264 22 9501 7 299 16 29 34 543 314
 901 418 4895 126 35 60]
label distribution of training set:
[ 2 815 12 2470 790 86 45 35 10 197 96 17 388 17
 22 25 14 294 3667 18 6698 2 207 10 20 21 367 225
 640 294 3384 89 28 38]
label distribution of val. set:
[ 0 85 2 363 116 16 8 6 5 35 18 2 70 2 6 2 4 40
 521 1 959 3 39 4 4 4 57 25 90 59 482 11 2 11]
label distribution of test set:
[ 1 224 2 646 218 18 11 11 1 54 28 6 93 6
 7 5 5 80 1076 3 1844 2 53 2 5 9 119 64
 171 65 1029 26 5 11]
```

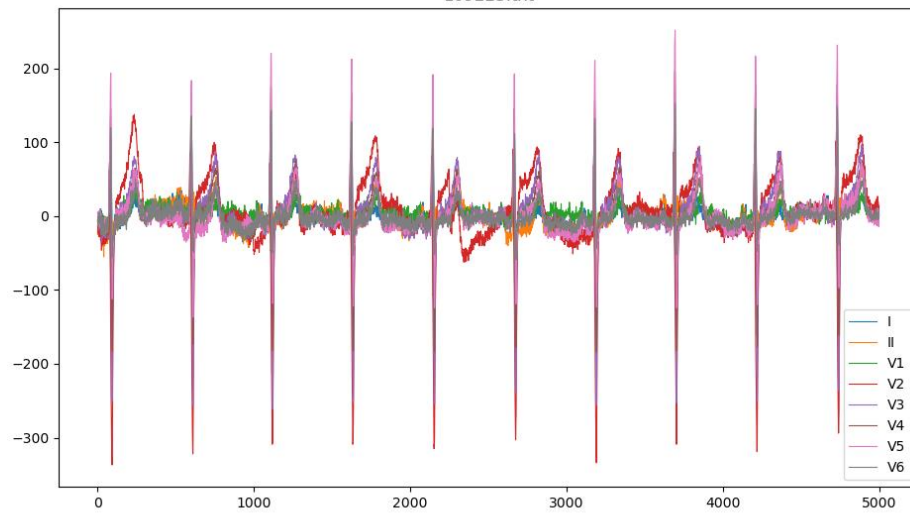


随机选取一个心电图样本可视化如下，不同的导联数据节拍一致，尖峰位置一致，各个导联间的数据有很强的相似性：

109223.txt



109223.txt



## 2 构建 Dataset

这部分代码见 `dataset.py`。

数据增强采用了小尺度的随机缩放,上下平移这两种添加噪声的方法。没有进行归一化。此外,最初还采用了竖直翻转,后来通过再次观察发现,34类异常中有一些“对称反相”的异常,比如说“电轴左偏”“电轴右偏”,“顺钟向转位”“逆钟向转位”,考虑到这一点,竖直翻转反而可能会误导模型,所以不使用竖直翻转。

自定义的 `dataset` 中 `__getitem__()` 返回的数据是: `x(8*5000)`, `age(1)`, `gender(1)`, `target(34, in one-hot)`。

## 3 模型

### 3.1 基于卷积的模型

说明：仅输入 8 导联的心电数据，未使用 age 和 gender 数据时，模型名称后加 Basic 作为标识。外加使用 age 和 gender 数据，方法是将其 concatenate 到网络的最后一层 FC 之前，模型名称后面没有加 Basic。

#### 3.1.1 ResNet

ResNet 是首先尝试的模型，输入的心电数据，可以将每个导联看成是一个通道，在保留 ResNet 整体结构的基础上，将 2D 卷积和池化都修改成了对应的 1D 版本，调整了部分参数，增大了各个 Conv1d 的卷积核大小，以适应较长的输入心电数据；在每个 Block 中均加入了一定的 dropout 来减小过拟合。

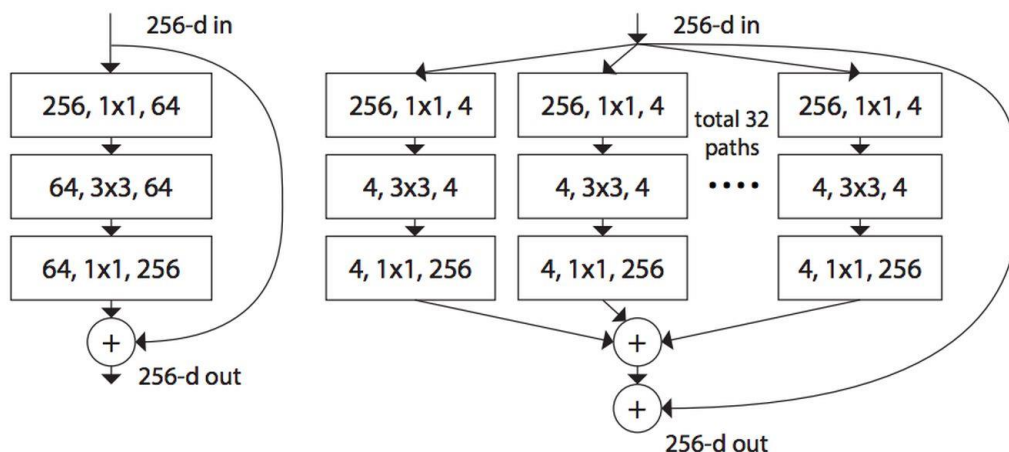
尝试了 ResNet-50 和 ResNet-101。

#### 3.1.2 ResNeXt

ResNeXt 在 ResNet 的基础上，吸收了 Inception 中各个 block 设计的 split-transform-merge 思想，将单路卷积变成多个支路的多路卷积，结构一致，进行分组卷积。

尝试了 ResNeXt-50-2x64d 和 ResNeXt-101-4x64d。

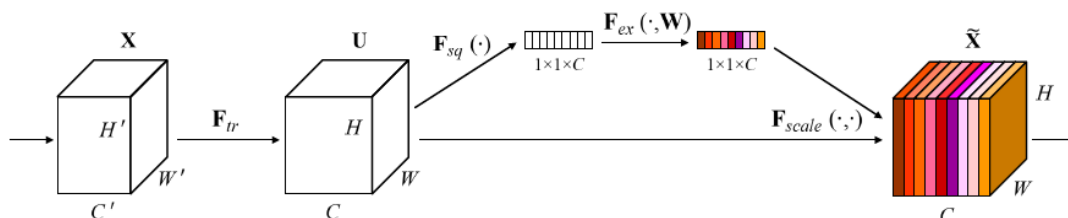
下图中，左侧：ResNet，右侧：ResNeXt。



### 3.1.3 SEResNet

SEResNet 在 ResNet 的基础上，对 channel 维度来进行特征压缩，学习不同 channel 的 scale factor，通过乘法逐通道加权到先前的特征上，完成在通道维度上的对原始特征的重标定。可以看成是在卷积网络中增加了 channel-wise attention。

尝试了 SEResNet-50 和 SEResNet-101。



### 3.1.4 ECGNet

再次观察可视化的结果，可以发现：将每个导联视作单个通道，在 backbone 中将 2D 卷积替换成 1D 卷积的想法十分自然，但是没有显式地利用各个导联之间的相似性；

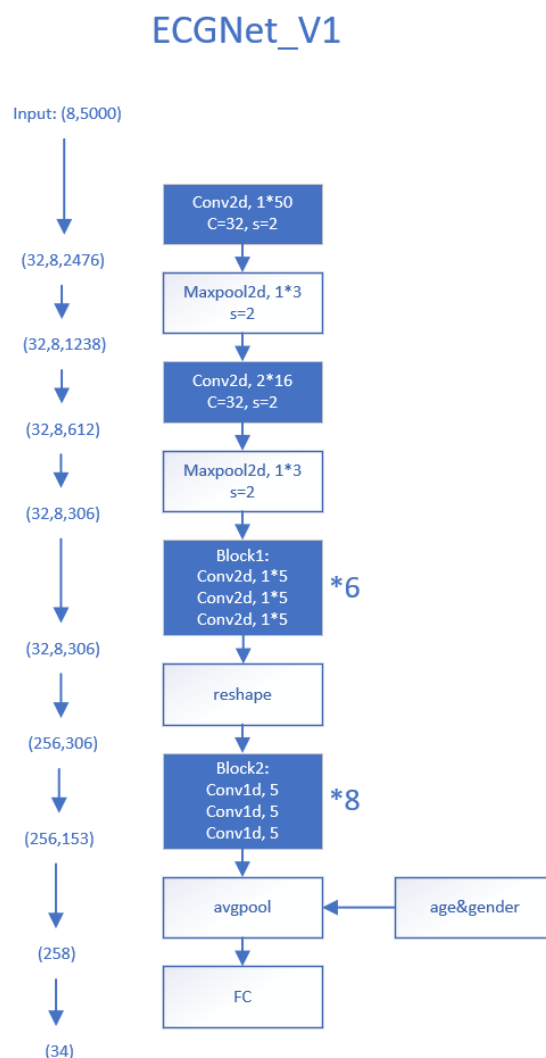
并且对于一维的导联数据，以前文可视化的例子为例，其整体的特征（在较大的长度跨度上）非常明显，在长度为 5000 的数据中有约 10 次的峰值；如果仍沿用 ResNet 中各部分均使用较小的卷积核（实际上在前文的模型中，卷积核的尺寸已经增大到 10 左右），恐怕难以较好地学习到这种整体特征；

考虑到这两点，对网络结构进行了改进，称为 ECGNet，如下（共有三个版本）：

ECGNet 网络仍采用基于 SEResNet 的结构实现，不同点是：第一点，为利用不同导联之间的相似性，将  $8 \times 5000$  的心电数据当作二维的数据处理，但是直接使用二维的卷积核也不合适，所以采用每个导联使用相同的卷积核，相比于直接使用一维卷积，网络增加了额外的维度；经过一定数量的二维卷积之后，将 channel 数和 8 导联数这两个维度合并，再进行一维卷积并回归；第二点是，在最初使用更长的卷积核（50）初步提取特征。

设计的 ECGNet 的结构如下，此为 ECGNet\_V1：

其中 Block1 是 3 次 2D 卷积，对 8 导联使用相同的卷积核；Block1 是 3 次 1D 卷积；两种 Block 在最后都进行 SE 和残差连接。

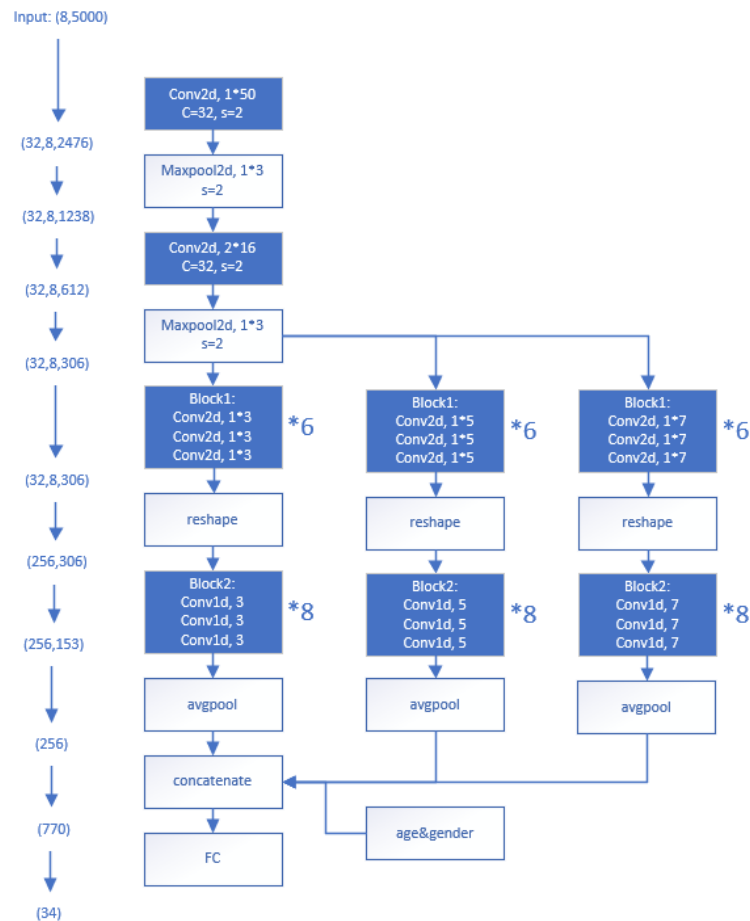


观察结构进一步考虑可以发现，两组 Block 中，卷积核的大小设定为 5，有一定的随意性，可否多试几种卷积核的大小，不同大小的卷积核对应着不同的感受野，将其 concatenate

到一起？

进一步改进，在 ECGNet\_V2 中，尝试了 3，5，7 三种卷积核大小并拼接到一起。

## ECGNet\_V2



检查 ECGNet\_V1 和 ECGNet\_V2，发现在平均池化 avgpool 之前的 Block2 仅做了 1 次下采样，输入的维数 306，输出的维数 153，对 153 维的输出做平均池化，这意味着 Block2 的卷积基本没有起到效果。

在 ECGNet\_V3 中，有四点改进：

在 Block2 处做了 4 次降采样，这与 ResNet 的做法保持一致；

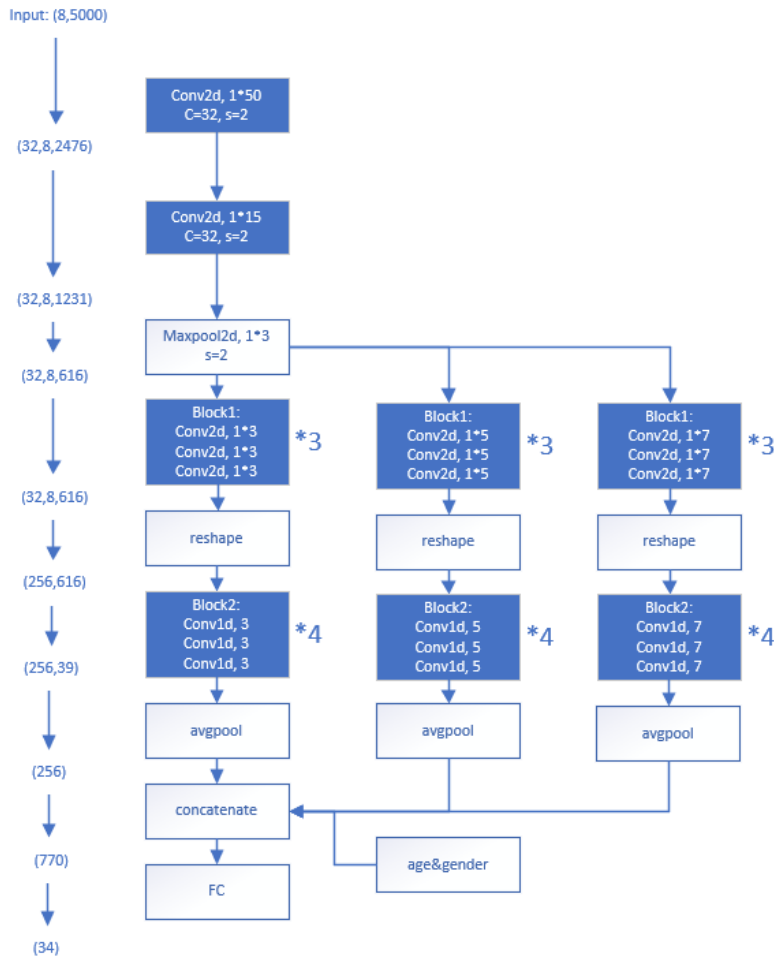
此外，在最初的卷积部分，由“卷积-池化-卷积-池化”的结构更改为“卷积-卷积-池化”的结构；

减小了堆叠的 Block 个数；

在 maxpool 之后额外增加了 dropout；



## ECGNet\_V3

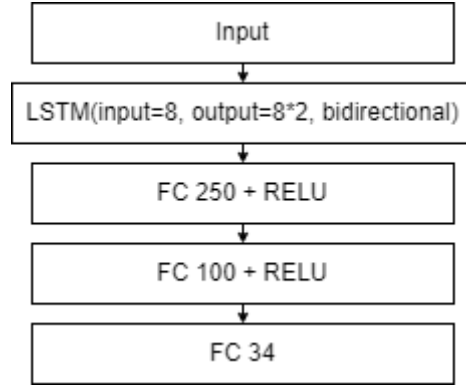


## 3.2 基于序列的模型

### 3.2.1 LSTM\_V1

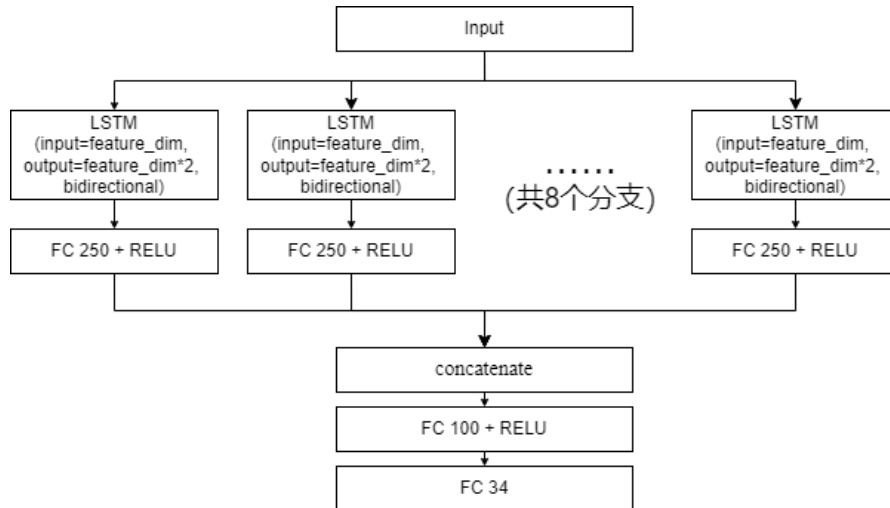
由于心电信号的特征与时间序列具有一定的关联，因此，尝试使用基于时序的 RNN 网络进行训练。考虑到心电信号中多次心跳信号之间可能存在的差异性对其特征的辨别，故首先使用可解决长期依赖问题的 LSTM 网络。

具体网络结构如下图所示。为增加心电信号中时序特征的提取，选择双向 LSTM 结构。其中，输入数据为 8 导联的心电数据，即数据列的时间长度为 5000，每个时间点的数据维度为 8。未使用到 age 和 gender 数据。



### 3.2.2 基于 8 分支的 LSTM\_V2

基于上述 LSTM 模型及其结果，将 LSTM 层的结构由单一支路更改为 8 个支路，结构一致，但模型参数不共享。每个支路的输入分别为 8 个导联中 1 个导联的心电数据，并将数据按时间顺序分为若干段，默认为 10 组数据，每组数据中含有 500 个数据点，即数据长度为 10，特征维度为 500 维。将 8 个支路的输出通过 concatenate 输入到网络的两层 FC 层，得到最终的分类结果。具体网络结构如下图所示。

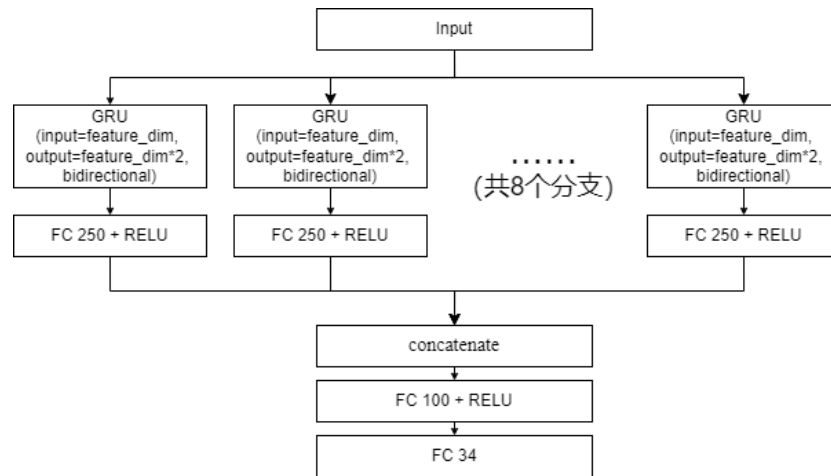


### 3.2.3 基于 8 分支的 GRU

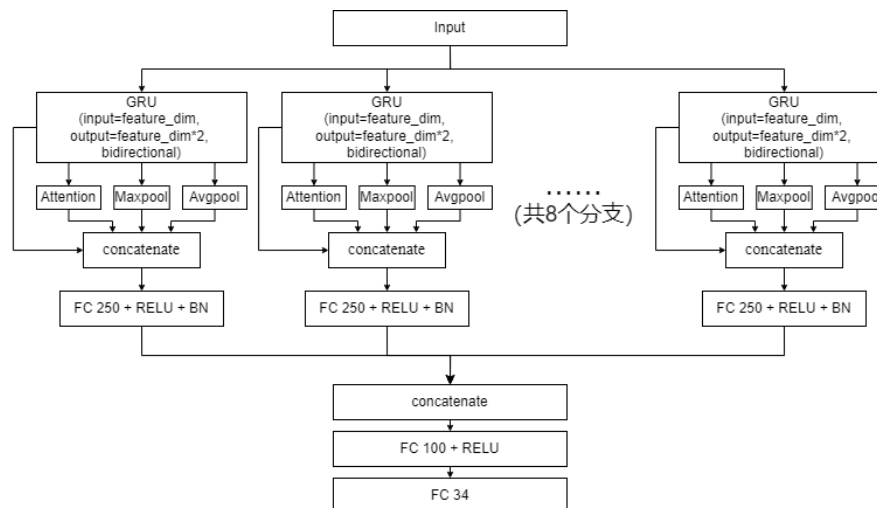
GRU 结构是 LSTM 的一个简化变体，其将 LSTM 结构中的忘记门和输入门合成为了一个更新门，并混合了细胞状态和隐藏层状态，使得模型比标准的 LSTM 模型要简单一些。

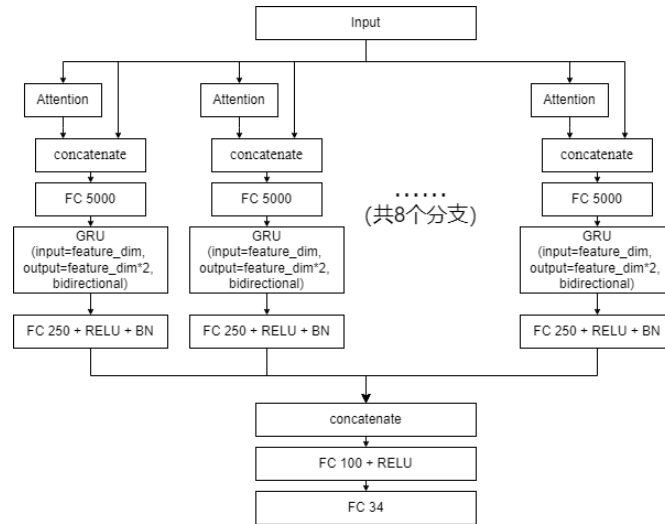
将上述模型中的 LSTM 层替换为 GRU 层，其余网络结构不变，记为 GRU\_V1。考虑到

8 导联之间的不同性与相似性，尝试分别采用 8 个分支采取同一 GRU 模型与不同 GRU 模型的两方案。具体网络结构如下图所示。



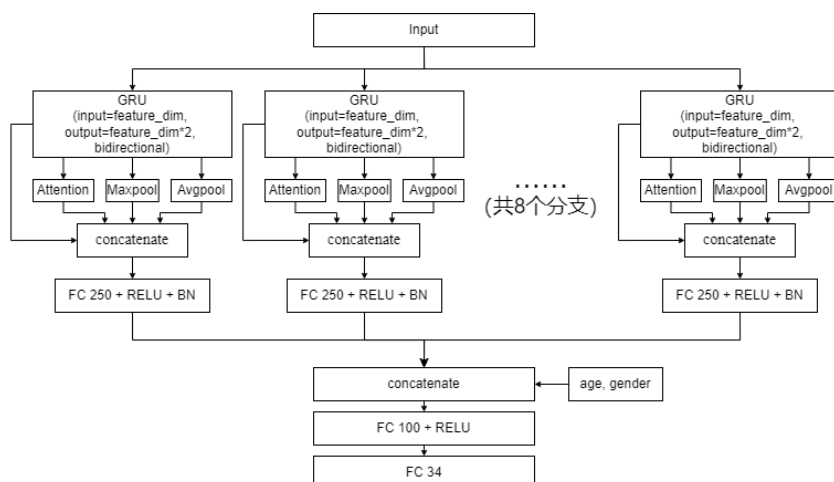
在心电数据中，不同部分对于最终标签分类的重要性有所不同，因此考虑引进 Attention 机制，给心电数据中对应心跳不同特征部分添加注意力权重，进而提升最终分类的正确率。在上述模型的基础上，考虑了两种引入 Attention 机制的方法，即在 GRU 层之后添加（GRU\_V2.1）和 GRU 层之前（GRU\_V2.2）添加，并分别进行训练尝试，其网络结构如下图所示。





添加 Attention 机制时，采用 Attention 层输出的 weight 信息与输入 Attention 层的数据进行 concatenate 后输入一个全连接层进行处理。与直接将其相乘比较，这样能较好地减少数据信息的损失，以充分提取心电数据中的特征。同时，将输入 Attention 层的数据分别通过平均池化 Avgpooling 和最大池化 Maxpooling 后的输出也输入至 concatenate 中，为提取特征提供更多的参考信息。

为充分利用所提供的心电数据，考虑在输入的数据中加入 age 和 gender 数据。而 age 和 gender 数据不属于序列信息，不需要 GRU 层的处理，因此，在上述模型基础上，在最后两层 FC 层之前 concatenate 输入数据中的 age 和 gender 信息，为标签的分类提供额外的信息。改进后的网络结构如下图所示（GRU\_V3）。



根据训练结果对网络中的部分参数进行调整：

(1) 输入心电数据的分段。分别将输入的时间长度为 5000 的心电数据分割为 500\*10、100\*50 和 50\*100 (feature\_dim\*time\_len)，探究更优的分割方法。

(2) GRU 层的 dropout。分别设置 dropout=0.3、0.1 进行训练。

(3) 各全连接层的输入与输出特征数。根据训练得到的分类结果准确率调整各连接层的输出特征数，最终结果如模型代码所示。

### 3.3 基于卷积和序列的模型

由于心电数据中同时包含有空间特征和时序特征，因此考虑将基于卷积的模型和基于序列的模型相结合，从而提取心电数据中的空间和时序特征，提高分类的准确率。

模型中的卷积网络部分采用 SEResNet50，时序网络部分采用添加有 Attention 机制的 GRU 网络。将原始心电数据输入 SEResNet50，将其最后一层 FC 前得到的结果直接输入 GRU 网络，进行时序特征的提取与分类。根据上述的基于时序特征的模型，可知基于 8 分支的 GRU 模型对心电数据的分类效果更好，因此，对 SEResNet50 网络进行以下 3 种适应性更改，再接入 GRU 网络进行训练：

(1) 设置 SEResNet50 网络输出时的 Avgpool 层 channel=1，并将其直接接入一个 GRU 通道的网络；

(2) 设置 SEResNet50 网络输出时的 Avgpool 层 channel=8，并将其直接接入具有 8 个支路的 GRU 通道的网络；

(3) 设置 SEResNet50 网络输出时的 Avgpool 层 channel=1，将输出数据重新排列为 8 个通道的数据后，接入具有 8 个支路的 GRU 通道的网络。

## 4 训练，验证和测试

这部分代码见 `main.py`。

要单独说明的是：

### 1) loss:

多标签分类问题，可以看作是对 34 个 label 进行二分类，使用的损失函数应该是二分类交叉熵损失函数（BCELoss），在数据准备中已经看到各个 label 对应的数据量差距很大，应当增加权重这一变量。所以使用

`torch.nn.BCEWithLogitsLoss`，参考[这里](#)，它是 sigmoid layer+BCELoss+weights。

其中  $weights \text{ for label } i = 1 / \ln(\text{data nums for label } i)$ ，在构建 Dataset 时完成了对 weights 的计算。

### 2) 优化器:

选用 Adam，默认参数。

### 3) lr\_scheduler:

首先尝试了分段常数衰减，需要在 config 里面设置：

而后尝试了 pytorch 库中的自适应调整 learning rate 函数 `torch.optim.lr_scheduler.ReduceLROnPlateau`，设置其 learning rate 变化的触发变量为 val F1。由于该函数的使用需要更改 `main.py` 中的函数，因此，在文件 `main_lr_opim.py` 中应用该 learning rate 调整函数。

### 4) 评价指标:

评价依据训练心电数据的真实标签以及网络预测得到的标签进行比较所得。其中，网络预测得到的标签由网络计算得到的每个标签的可能性与阈值判断得到。

F1 score、Precision（精确率）、Recall（召回率）：由 `sklearn.metrics` 库中的 `precision_recall_fscore_support` 函数，根据心电数据的真实标签和预测标签比较所得。

Acc（准确率）：对于每份心电数据，比较其真实标签与预测标签，当且仅当该份数据的真实标签与预测标签完全吻合的时候判断该份数据预测准确。每个 epoch 的  $Acc = \text{预测准确的数据份数} / \text{batch\_size}$ 。

### 5) 其它技巧:

early stopping：每训练一个 epoch，在验证集上测试一次，更新验证集上 F1 score 最高的模型权重。

## 5 结果

### 5.1 基于卷积的模型

Model	test F1	train F1	val. F1	lr	train epoch	Acc.	Precision	Recall
ResNet-50-Basic	0.882	0.883	0.884	1e-4	50	NA	NA	NA
ResNet-101-Basic	0.883	0.890	0.885	5e-5	200	NA	NA	NA
ResNeXt-50-Basic	0.856	0.863	0.858	5e-3	120	NA	NA	NA
ResNeXt-101-Basic	0.875	0.900	0.870	5e-3	200	NA	NA	NA
ResNet-50	0.898	0.936	0.903	5e-3	200	0.785	0.959	0.844
ResNet-50	0.906	0.975	0.909	1e-4	200	0.803	0.938	0.877
ResNet-50	0.903	0.966	0.905	5e-5	200	0.793	0.944	0.867
SEResNet-50	0.914	0.975	0.916	1e-4	200	0.815	0.949	0.881
SEResNet-101	0.908	0.950	0.892	1e-4	200	0.802	0.952	0.869
ECGNet_V1	0.912	0.985	0.911	1e-4	200	0.814	0.950	0.878
ECGNet_V1	0.913	0.970	0.910	5e-5	200	0.815	0.953	0.876
ECGNet_V2	0.915	0.987	0.911	1e-4	200	0.817	0.951	0.882
ECGNet_V2	0.911	0.975	0.906	5e-5	200	0.811	0.952	0.874
ECGNet_V3	0.915	0.995	0.912	1e-4	200	0.817	0.946	0.886
ECGNet_V3	0.907	0.961	0.906	5e-5	200	0.803	0.957	0.863

备注：1.train F1, val. F1均取最后一个epoch的数据(last)，test F1取best和last测试中较高者；  
2.train epoch=200，在epoch为50，100，150降低学习率为之前的十分之一，epoch=50/120时类似；  
3.Acc., Precision, Recall是测试集的结果，删去了一开始错误的方法得到的结果，即表中前几行的NA。

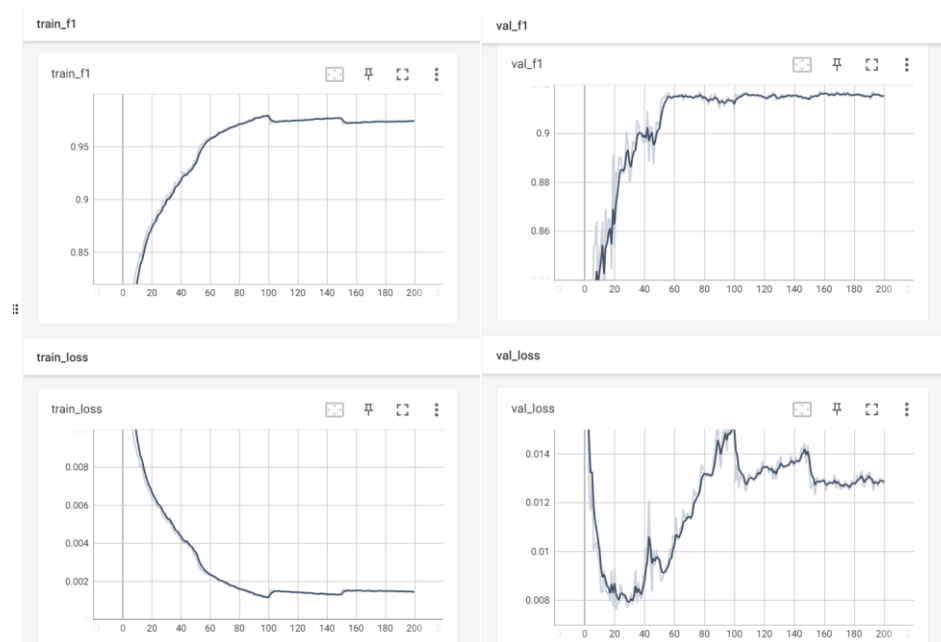
test F1 最高的是 ECGNet\_V3，最高为 0.915，其次是 ECGNet\_V2 和 SEResNet-50；

test Acc.最高的是 ECGNet\_V3，最高为 0.817。

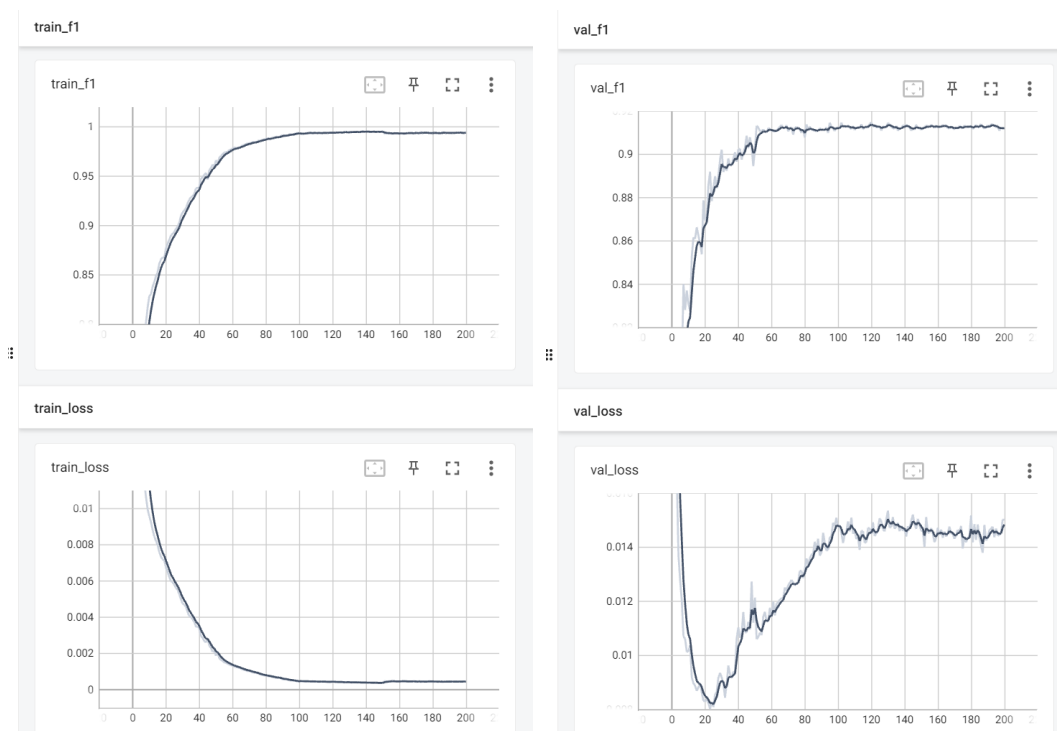
综合来看效果最好的是 ECGNet\_V3。在多项指标上达到最高。

两个典型例子：

SEResNet-50 训练过程中 train F1, train loss, val.F1, val. loss 曲线记录如下：



ECGNet\_V3(lr=1e-4) 训练过程中 train F1, train loss, val.F1, val. loss 曲线记录如下：



可见，train F1, train loss, val.F1 的变化曲线比较正常，val. loss 的变化曲线有些奇怪，可能是训练集与测试集不匹配，详见 5.4 结果分析一节。

## 5.2 基于序列的模型

Model		fea_dim*time_step	test F1	train F1	val. F1	train epoch	Acc.	Precision	Recall	lr	dropout
LSTM_V1		8*5000	0.000	0.000	0.000	256	0.000	0.000	0.000	分段lr	0.3
LSTM_V2		500*10	0.846	0.965	0.849	200	0.707	0.946	0.765	分段lr	0.3
GRU_V1	8 gru	500*10	0.863	0.996	0.859	256	0.732	0.937	0.800	分段lr	0.3
	same gru	500*10	0.861	0.996	0.853	200	0.725	0.932	0.801	分段lr	0.3
GRU_V2.1		500*10	0.872	0.913	0.875	256	0.747	0.963	0.796	分段lr	0.3
GRU_V2.2		500*10	0.737	0.957	0.735	256	0.560	0.895	0.627	分段lr	0.3
GRU_V3		500*10	0.879	0.961	0.880	256	0.755	0.953	0.815	分段lr	0.3
		100*50	0.875	0.957	0.875	256	0.753	0.953	0.809	分段lr	0.3
		50*100	0.870	0.925	0.872	256	0.738	0.944	0.807	分段lr	0.3
		50*100	0.871	0.936	0.875	256	0.747	0.953	0.803	自适应lr	0.3
		50*100	0.885	0.981	0.883	256	0.764	0.940	0.836	自适应lr	0.1

备注：1. 分段lr的参数为：第32个epoch之前为0.001，第64个epoch之前为1e-4，第128个epoch之前为1e-5，第256个epoch之前为1e-6；  
 自适应lr的参数为：以val. F1作为敏感变量，初始lr为0.001，每当10个epoch内val. F1不再上升，则lr缩小10倍，直至1e-8。  
 2. 各评价指标均为最后一个epoch训练结束时得到的指标，其中，Acc、Precision、Recall均为测试集计算得到的评价指标

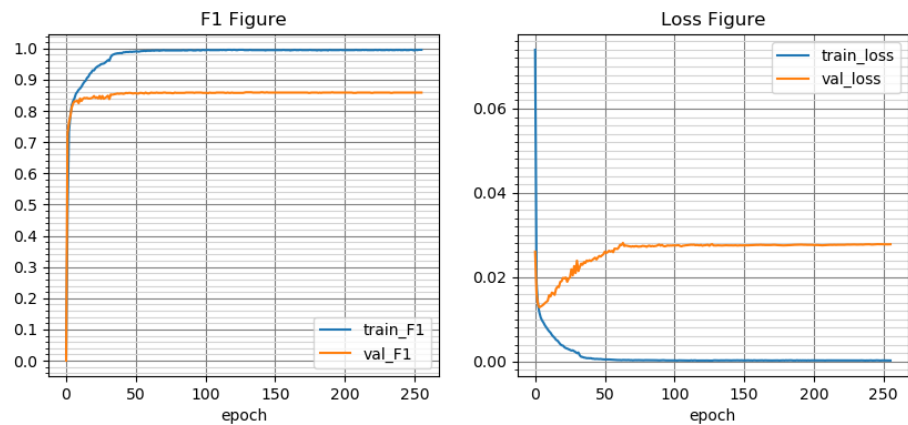
test F1 最高的是 GRU\_V3 (dropout=0.1, fea\_dim\*time\_step=50\*100, 自适应 lr)，最高为 0.885；test Acc.最高的是 GRU\_V3，最高为 0.764。

由于模型训练有时存在一定的随机性，因此综合多次训练结果来看，效果最好的是 GRU\_V3，在多项指标上达到最高。

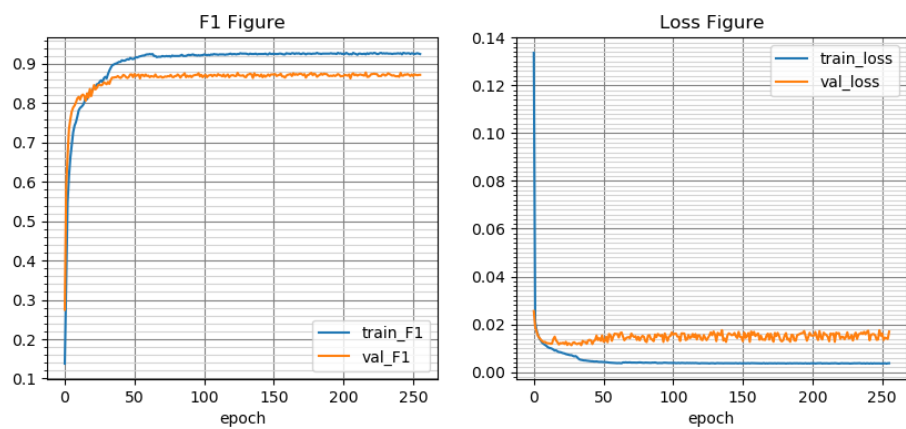
三个典型的例子：

GRU\_V1(8 gru)训练过程中 train F1, train loss, val.F1, val. loss 曲线记录如下：

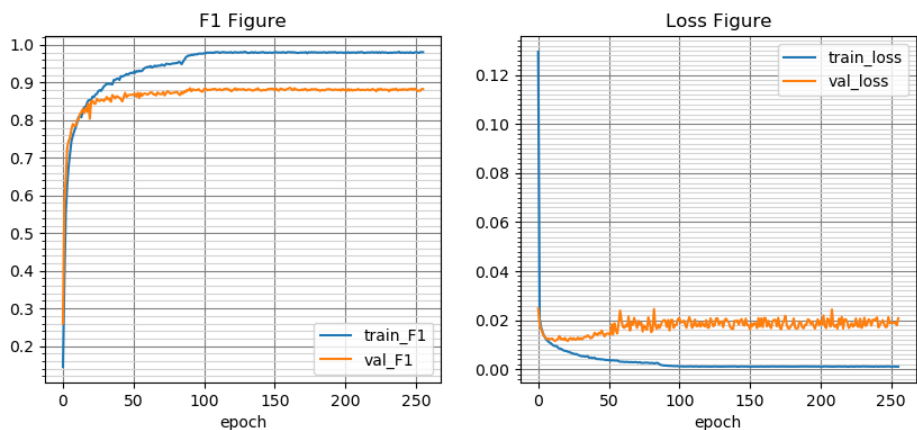




GRU\_V3(50\*100, 分段 lr)训练过程中 train F1, train loss, val.F1, val. loss 曲线记录如下:



GRU\_V3(50\*100, 自适应 lr, dropout=0.1)训练过程中 train F1, train loss, val.F1, val. loss 曲线记录如下:



有上图可知, 训练时的 loss 函数会在一定程度上影响训练的最终结果, 若验证集的 loss 函数过大, 可能会导致过拟合等问题, 使得最终分类准确率略低。

### 5.3 基于卷积和序列的模型

Model	test F1	train F1	val. F1	train epoch	Acc.	Precision	Recall
SEResNet_GRU_V1	0.908	0.906	0.955	256	0.806	0.960	0.862
SEResNet_GRU_V2	0.899	0.968	0.893	200	0.787	0.963	0.842
SEResNet_GRU_V3	0.914	0.965	0.914	265	0.815	0.957	0.875

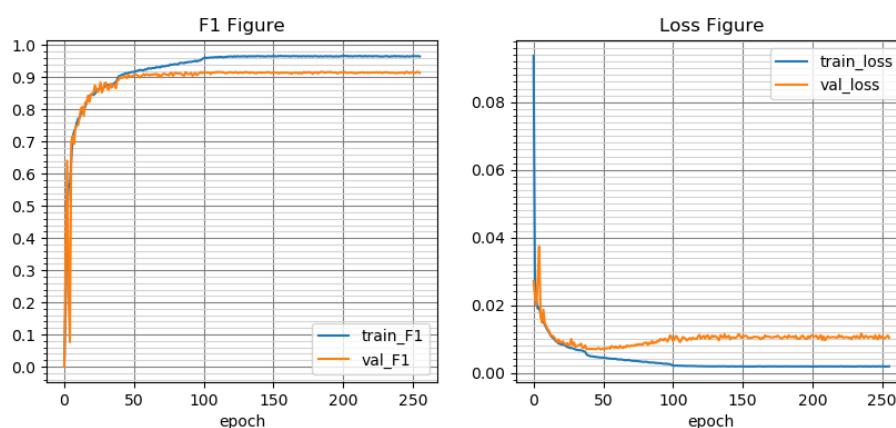
备注：1、lr采用自适应lr，其参数为：以val. F1作为敏感变量，初始lr为0.001，每当10个epoch内val. F1不再上升，则lr缩小10倍，直至1e-8。  
2、各评价指标均为最后一个epoch训练结束时得到的指标，其中，Acc、Precision、Recall均为测试集计算得到的评价指标

test F1 最高的是 SEResNet\_GRU\_V3，最高为 0.914；test Acc.最高的是 SEResNet\_GRU\_V3，最高为 0.815。

综合来看效果最好的是 SEResNet\_GRU\_V3，但与卷积模型与序列模型相比较，其效果好于序列模型，与其所基于的卷积模型相当。

一个典型的例子：

SEResNet\_GRU\_V3 训练过程中 train F1, train loss, val.F1, val. loss 曲线记录如下：



由图中可知，训练集与验证集的 F1 值的区别与第 50-100 个 epoch 时 loss 的变化存在一定的相关性，而 val\_loss 的变化与 SEResNet-50 模型中 val\_loss 的变化相似，可能是由于过拟合所导致。

### 5.4 结果分析

可以发现，几乎所有的模型中 train F1 要明显高于 val. F1 和 test F1，模型可能出现了过拟合现象；分析原因，划分数据集时，train:val.:test=7:1:2，测试集的比重应当更大一些；另外，由于部分的异常情况占比很少，对这类异常情况对应的样本划分时也不均匀，模型通过训练集学习到的“知识”与测试时有偏差，导致测试的效果与训练的效果有差异。

为了减小过拟合，在上文中已经提到的 **early stopping** 的基础上，在基于卷积的各模型中均加入了一定的 **dropout**。

另外，更复杂的模型效果反而没有较为简单的模型效果好，除了过拟合这个因素之外，也可能是数据本身没有需要很深的网络才能提取的特征？如果对心电数据有更清楚透彻的认识，可能会设计出更好的模型。

对于基于序列的模型，其最终效果不及基于卷积的模型，其原因可能心电信号本身的特征和模型结构两方面所导致。心电信号中的异常通常出现在整段信号的某个局部区域，空间特征比时序特征更显著，因此可能导致基于卷积的模型对特征的识别和提取更有效。其次，所设计的模型中，基于序列的模型均没有基于卷积的模型深，结构较为简单，可能导致没有完全提取数据中的特征信息，进而导致准确率的下降。

另外，基于序列的模型 **GRU\_V3** 与基于卷积的模型 **ResNet50** 所得到的 **test F1** 值相当，但在模型训练的时间上，**GRU\_V3** 需要约 90min，而 **ResNet50** 则需要多于 5h，其中一定原因是由于基序列的模型层数较少，训练效率较高，同时，也可能与心电信号的特征相关。

对于基于卷积和序列的模型，将卷积网络的输出作为 **GRU** 网络的输入，但最终得到的结果却与仅仅基于卷积的模型相关，或是比其更差。其原因可能与模型设计相关，即卷积网络与 **GRU** 网络的连接方式；同时，较深的卷积网络使得其输出数据中的时序特征减弱，导致后续接上的 **GRU** 网络无法进一步提取和识别心电数据中的特征，进而导致识别准确率的降低。

## 6 改进

1. 可以考虑手工提取一些传统特征，首先可以想到的是一些直接的统计特征（如均值，方差等）和频域信号，以及一些在实际衡量心电信号中的一些参数所表征的特征。这些特征与年龄和性别特征一起，与深度网络输出的数据一起进行回归；

2. 可以尝试进行多模型的融合，比如可以将不同的卷积模型的输出简单融合，如 ResNet50 与 ResNeXt50 最后一层的回归取平均作为输出特征；

3. 基于卷积的模型部分，在最初尝试进行训练的过程中，没有提前进行规划，各个模型的训练参数都有所不同；同时更换模型和更改参数，感觉这样做不太好.....后面训练的几个模型，有确定的 epoch 和 lr；此外，网络的输入数据，网络的选择和具体结构细节，优化器，损失函数，训练策略.....等各个部分都有继续斟酌和改进的空间；

4. 基于序列的模型部分，可以尝试不同的序列分割方式、更深一点的网络，以及更改一些网络的结构，使其更适于提取心电信号的特征；

5. 基于卷积和序列的模型部分，可以尝试不同的卷积网络与序列网络之间的结合、数据进入网络的顺序先后、卷积层与序列层如何更有效地连接等。

-----

感谢课程提供的服务器资源！