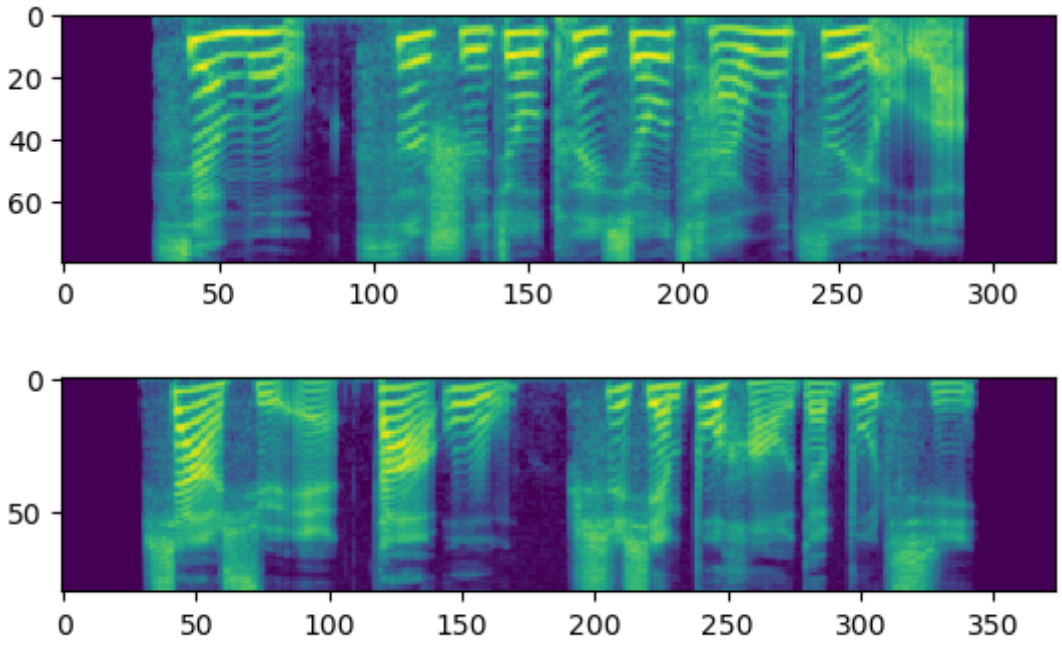
语种识别任务项目报告

23210980044 加兴华

问题分析

我们对训练数据中两类标签下的语音转梅尔频谱样本进行绘制观察，图像如下：



图像尺寸分别为(321, 80)和(374, 80)，这说明：第一个维度代表语音的时间步长，每个样本的长度不同；第二个维度代表梅尔频谱的频率特征，每个样本的特征数相同。

此外，训练数据的样本时间长度统计为

```
{'max_length': 734, 'min_length': 120, 'avg_length': 293.1455,
'percentile25_length': 231.0, 'percentile50_length': 281.0,
'percentile75_length': 344.0}
```

而测试数据的样本时间长度统计为

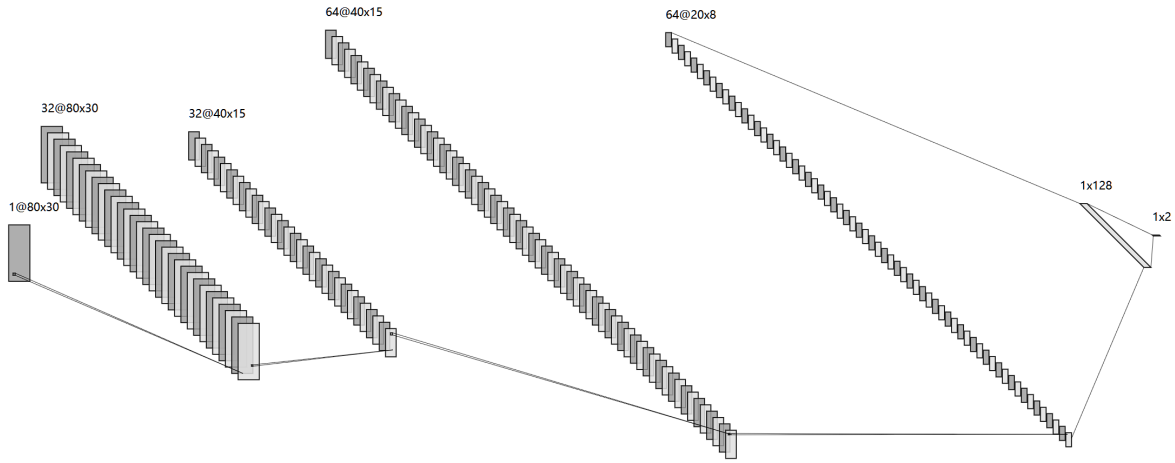
```
{'max_length': 766, 'min_length': 40, 'avg_length': 190.651,
'percentile25_length': 72.0, 'percentile50_length': 199.0, 'percentile75_length':
288.0}
```

这说明测试数据和训练数据存在一定的分布差异。

模型构建

CNN模型

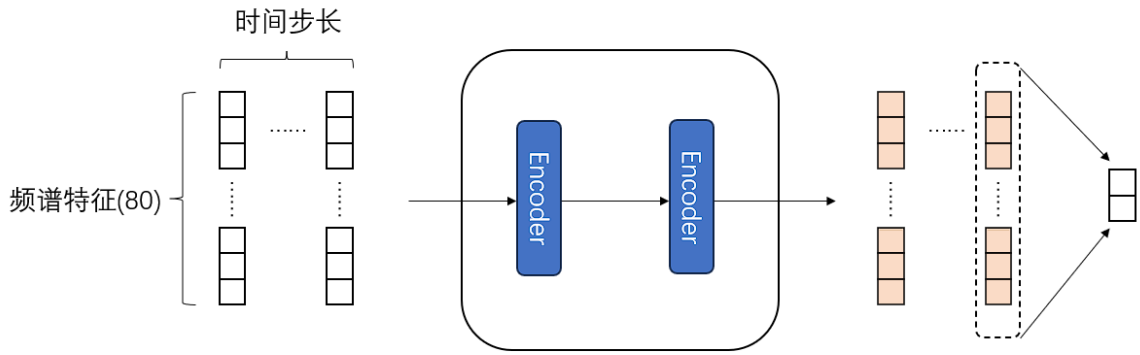
在这一部分，我将介绍使用的卷积神经网络（CNN）模型，包括模型架构、损失函数、优化器的选择等方面的内容。使用CNN模型的动机在于把语种识别任务视为一种图像分类任务，以图像为输入，类别为输出。我采用的CNN模型框架如下图所示：



模型首先通过3x3卷积将单通道图像转变为32通道特征，其余尺寸保持不变，接着通过池化将长度和宽度减半，然后再通过3x3卷积将特征转变为64通道特征，并再次池化，最后平铺经过一个128维的全连接层到达一个2维全连接层输出分类结果。损失函数选用交叉熵损失，优化器为初始学习率=5e-4的Adam优化器。

Transformer模型

在这一部分，我将介绍使用的Transformer模型。使用Transformer模型的动机在于把语种识别任务视为一种序列分类任务，以向量序列为输入，最后一步的预测为类别输出，模型框架如下图所示。



该模型以transformer模型的encoder块(2层4头encoder)作为主干，将提取特征的最后一个时间步传到一个2维全连接层输出分类结果。损失函数与优化器选择与前文CNN模型所用相同。

实验

数据预处理

基于数据观察，需要先对将变长数据处理成固定尺寸才能送入模型，主要操作为先截掉样本中左侧的空白部分，然后设定一个阈值，对时间步长大于阈值的样本执行截断操作；对时间步长小于于阈值执行零填充操作。

数据集划分

我将训练数据随机打乱后按8：2的比例划分获得训练集与验证集。

训练参数

在本项目实验中，输入样本batch size=64，训练轮数=20，所用设备为单张**GeForce RTX 3090**显卡。

实验分析

对两种模型在不同阈值水平上的训练结果如下：

Model	**Train Accuracy(%)**	**Valid Accuracy(%)**	**Valid Wrong Kurtosis**	**Valid Wrong Skewness**
CNN(threhold=30)	96.28	93	-0.0717	0.6652
Transformer(threhold=30)	90.5	87.5	1.049	1.056
CNN(threhold=300)	100	94	**-0.3004**	0.25181
Transformer(threhold=300)	91.81	92	2.008	1.188

其中，**Valid Wrong Kurtosis**与**Valid Wrong Skewness**分别为验证集中错分类样本的峰度和偏度。考虑到训练数据和测试数据存在分布差异（在问题分析一节中已经阐述），应该选用泛化性强的模型，而**Valid Wrong Kurtosis**值越低表明错误分类样本越均匀，从而模型在不同分布数据上的表现越一致。

综上，我最终选用了CNN模型，并设定阈值为300对其进行训练。

总结

在本项目中，我设计了CNN模型和Transformer模型来完成语种识别。在训练过程中，我观察到CNN模型收敛更快，最终验证精度更高，并且验证集上的错误样本分布更加均匀。因此，我认为语种识别任务更适合于图像分类而不是序列预测。